Matthew Bolding
Dr. Nelis Potgieter
MATH 40883–015
17:00 CDT, March 18th, 2022

Predictive Modeling: Project 1

1 Visualizing Data

Before visualizing any data, we must first read the data from a file.

```
survey <- read.table("am_com_survey.txt", header = T, sep = ",")
```

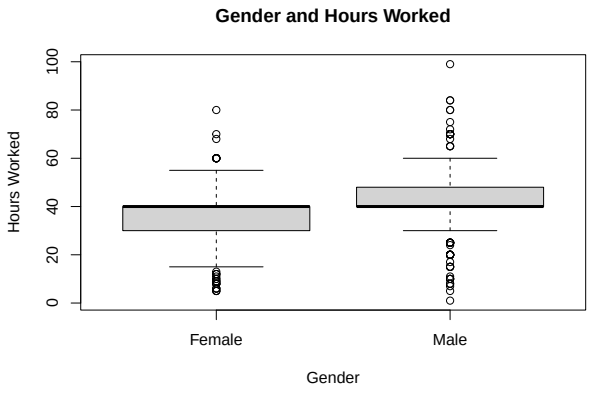
We may now begin to examine and visualize the relationships and interactions between predictors and the output variable. `log_inc`. However, before visualizing any data, determining the correlations between variables, via `round(cor(survey), 2)`, might provide an education starting point when determining which variables to display graphically.

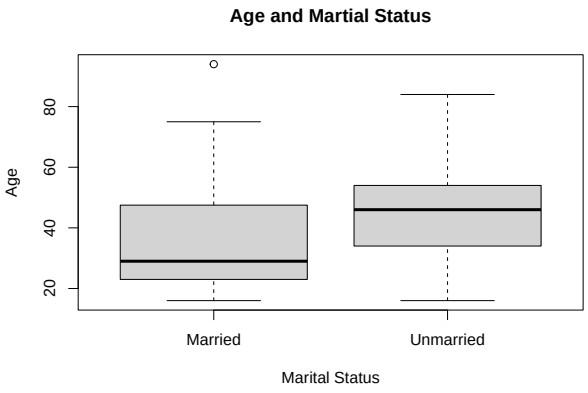
```
##              hrs_work  race   age gender time_to_work married   edu log_inc
## hrs_work         1.00 -0.05  0.11  -0.27         0.12    0.02 -0.12    0.61
## race            -0.05  1.00 -0.05   0.07         0.01   -0.03  0.08   -0.12
## age              0.11 -0.05  1.00  -0.02         0.03    0.27 -0.05    0.33
## gender          -0.27  0.07 -0.02   1.00        -0.06    0.44 -0.02   -0.26
## time_to_work     0.12  0.01  0.03  -0.06         1.00    0.04 -0.11    0.15
## married          0.02 -0.03  0.27   0.44         0.04    1.00 -0.06    0.14
## edu             -0.12  0.08 -0.05  -0.02        -0.11   -0.06  1.00   -0.30
## log_inc          0.61 -0.12  0.33  -0.26         0.15    0.14 -0.30    1.00
```

1.1 Visualizing Two Predictors

Let us first visualize predictor pairs `gender` and `hrs_work` as well as `age` and `married`.

```
female.hrs_work <- survey[which(survey$gender == 1), ]$hrs_work
male.hrs_work <- survey[which(survey$gender == 0), ]$hrs_work
boxplot(female.hrs_work, male.hrs_work,
        ylab = "Hours Worked",
        xlab = "Gender",
        names = c("Female", "Male"),
        main = "Gender and Hours Worked")
```



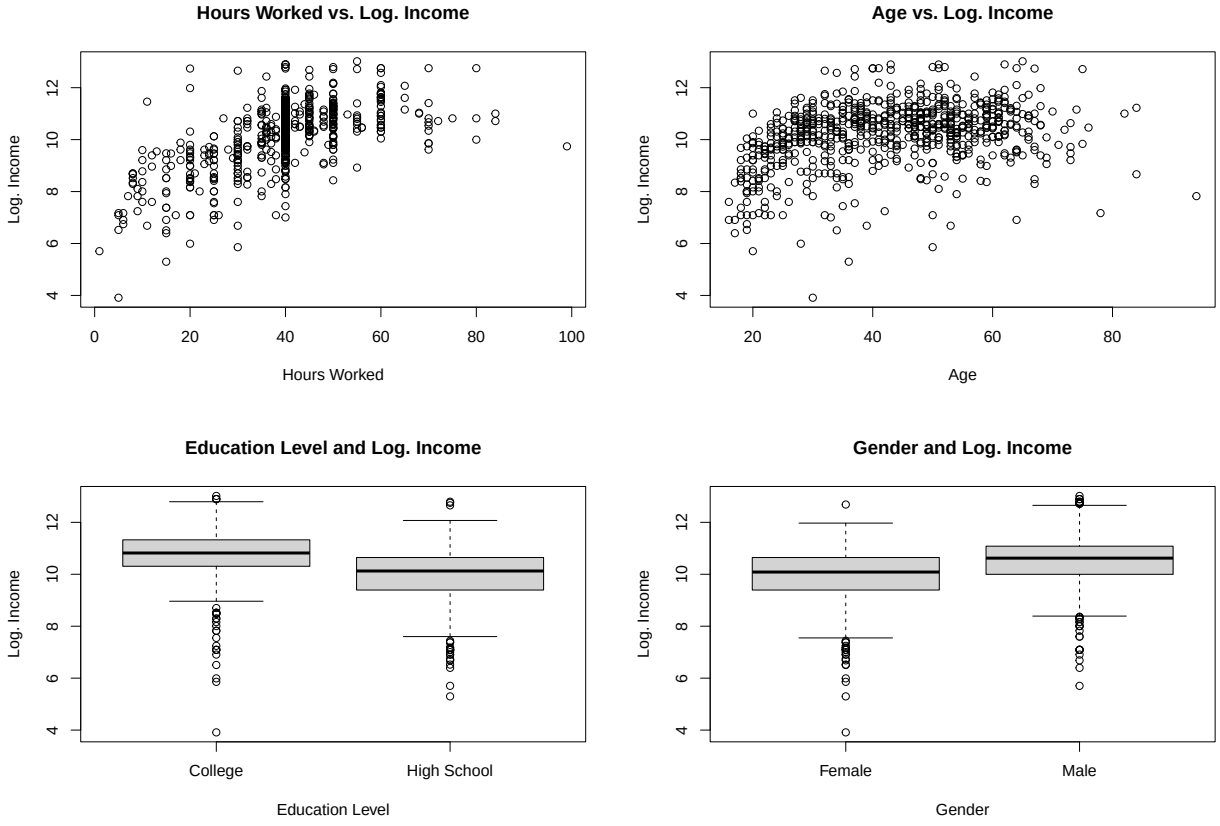


A similar block is used to visualize `age` and `married` but is not shown for brevity.

Note that `gender` and `hrs_work` have a correlation high enough (-0.27) so that there might be some separation between the two variables. Similarly, `age` and `married` have a relatively high correlation (0.27).

Although `gender` and `married` are predictors with the highest correlation (0.44), we will not display them since their graph—a pie chart or bar graph—shows little information in actuality. These predictors are further discussed in **Section 2—Exploring Relationships**.

1.2 Visualizing a Predictor and `log_inc`



In the above graphs, chosen based on the high(-ish) correlation between the predictor and the output variable, we see some form of a linear relationship with the output variable. These plots were constructed with a simple `plot()` function; code can be found in the `.rmd` file or standalone `R` file.

2 Exploring Relationships

Consider the plots in **Section 1.2**. Observe that they all exhibit some linear relationship to the output variable, no matter how weak. These predictors, therefore, would likely contribute well.

Recall the explicitly mentioned correlations in **Section 1—Visualizing Data**. As we saw in the above section, some predictors, like `gender` and `hrs_work` as well as `age` and `married` have strong enough correlations to be of concern.

These correlations might present an issue when constructing a model. Determining the exact effect a predictor has on the output variable might be difficult—separating the effects of two such predictors that carry a relatively high correlation presents a hefty challenge. Colinearlity—caused by a high correlation—can increase the standard error of predicted coefficients. Therefore, it might be wise to introduce an interation term to better capture the effects the two predictors have on the output variable.

Referencing the *Gender and Hours Worked* graph above, the box plot visually shows the aforementioned separation, motivated by the moderately high correlation. The same can be said for the `age` and `married` predictors but to a lesser extent—their box plots still have some overlap but significantly less than the previous pair of predictors. So, if we are to construct a linear model contains both either of these pairs of preditors, an interaction term might increase the model's accuracy.

Visualizing `gender` and `married` would show that there is not a single observation of an unmarried woman. In fact, most observations are of married females. (This can be verified by counting the number of rows for all combinations of `gender` and `married` using the `which` command.) This is an interesting characteristic of the data set, though I'm unsure how it would affect the resulting models. Perhaps predicting the income for an unmarried female would be a leverage value.

3 Single Qualitative and Quantitative Models

3.1 `hrs_work` and `gender`

These two predictors make good candidates for a linear regression model with a single qualitative and quantitative input variable. Not only does `hrs_work` have the highest correlation to the output variable, but this input variable and `gender` have a rather high correlation, as we discussed earlier. `gender` has some correlation to the `log_inc`, too. As such, we see below that the interaction term between the two variables is statistically significant.

To train the model, we run the below command. (Note: only viewing the fourth column of the coefficients from `summary` will display the p-values, and a colon between two input variables will form an interaction term.)

```
model.hrs_work.gender <- lm(log_inc ~ hrs_work + gender + hrs_work:gender, data = survey)
summary(model.hrs_work.gender)$coefficients[,4]
```

```
##     (Intercept)        hrs_work          gender hrs_work:gender
##   8.754964e-240    2.692876e-33    3.310487e-05    1.164705e-03
```

3.2 `age` and `edu`

The next model again includes `age` and `edu` for the same reason as above—they both have a high correlation to the output variable. (See the visualizations above.) However, the correlation between these two inputs is rather low. Consequently, after training the model, we find that the p-value of the interaction term greater than the significance level; hence, it's removed. It's not shown, but the model without the interaction term has `age`'s and `edu`'s p-values retaining their significance.

```
model.age.edu <- lm(log_inc ~ age + edu + age:edu, data = survey)
summary(model.age.edu)$coefficients[,4]
```

```
##   (Intercept)           age           edu       age:edu
## 2.664882e-214  2.497510e-07  1.439345e-03  5.507931e-01
```

3.3 `age` and `married`

This choice of selecting `age` stems from its correlation to the output variable. Due to the constraints of the project, we may only pair a quantitative predictor, such as `age`, with a categorical variable. `married` has the next highest correlation to the output variable among unused categorical variables paired with `age`.

```
model.age.married <- lm(log_inc ~ age + married + age:married, data = survey)
summary(model.age.married)$coefficients[,4]
```

```
##   (Intercept)           age       married   age:married
## 2.077849e-168  1.182601e-08  2.677694e-02  8.582655e-02
```

We see that the interaction term is not statistically significant, so we will discard that term from the model, which is retrained without it. (Note: after the removal of the interaction term, `married` becomes statistically insignificant!)

4 Backwards Selection

The backwards selection process dictates that the starting point for the eventual model is the saturated one—a model using all predictors. The process further instructs us to remove the predictor with the largest p-value.

```
model.back.sel <- lm(log_inc ~ hrs_work + race + age + gender + time_to_work +
                               married + edu, data = survey)
p_values <- summary(model.back.sel)$coefficients[ ,4]
p_values[p_values == max(p_values)]
```

```
## time_to_work
##      0.13573
```

Such a predictor with the largest p-value would be `time_to_work`. Now, we retrain the model, like above, but without that predictor. The largest p-value of the newly-trained model is below.

```
p_values <- summary(model.back.sel)$coefficients[ ,4]
p_values[p_values == max(p_values)]
```

```
##       race
## 0.06175867
```

Repeating this process, we remove the predictor `race`, since it has 1) the largest p-value, which is 2) greater than the significance level. Then, we retrain without `race`.

```
model.back.sel <- lm(log_inc ~ hrs_work + age + gender + married + edu, data = survey)
p_values <- summary(model.back.sel)$coefficients[ ,4]
p_values[p_values == max(p_values)]
```

```
##      married
## 1.871347e-06
```

Now, we do not remove the least statistically significant predictor, since all inputs variables have a p-value less than the standard significance level of 0.05. Therefore, we have completed the backwards selection process to generate a model.

5 Estimated Prediction Equations

5.1 Backwards Selection

After getting the coefficients (via `round(coef(model.back.sel), 4)`, not shown for brevity), we can determine the model's prediction equation to be

$$\begin{aligned}\widehat{\texttt{log\_inc}} = 7.6534 &+ (0.0491 \times \texttt{hrs\_work}) + (0.0187 \times \texttt{age}) \\ &+ (-0.4633 \times \texttt{gender}) + (0.4418 \times \texttt{married}) + (-0.5394 \times \texttt{edu}).\end{aligned}$$

All subsequent model equations' coefficients are queried in the same fashion. See hidden code in `project1.rmd`.

5.2 hrs_work and gender

$$\widehat{\texttt{log\_inc}} = 8.4355 + (0.0486 \times \texttt{hrs\_work}) + (-0.9994 \times \texttt{gender}) + (0.0194 \times (\texttt{hrs\_work} \times \texttt{gender})).$$

5.3 age and edu

$$\widehat{\texttt{log\_inc}} = 9.4888 + (0.0265 \times \texttt{age}) + (-0.6900 \times \texttt{edu}).$$

5.4 age and married

$$\widehat{\texttt{log\_inc}} = 8.9349 + (0.0263 \times \texttt{age}) + (0.1789 \times \texttt{married}).$$

5.5 Recommended Model

Adjusted R^2 is a good measure to determine how well a model fits the variability of the data set upon which it's trained. The adjusted R^2 of ...

- `model.hrs_work.gender` is 0.3833474,
- `model.age.edu` is 0.1861216,
- `model.age.married` is 0.1112636, and
- `model.back.sel` is 0.5097043.

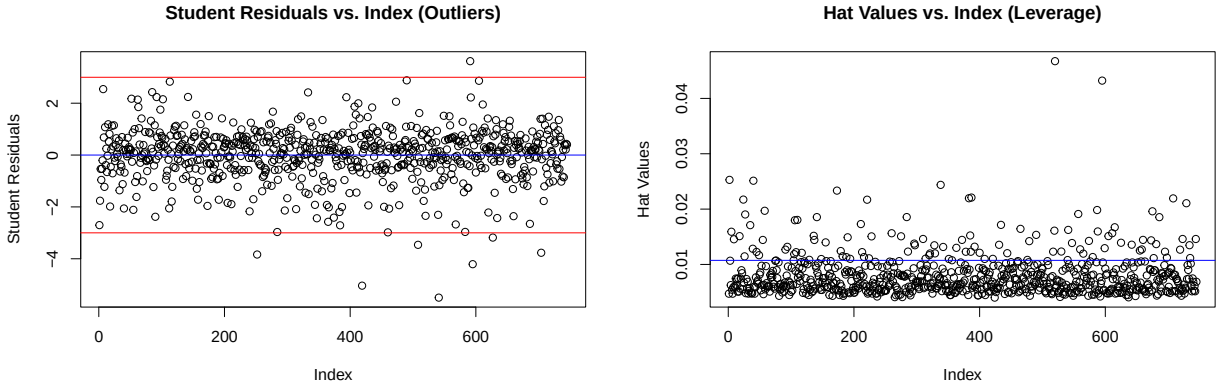
Hence, we choose the backwards selection model, since it has the highest adjusted R^2 value.

6 Outliers and Leverage Values

Via the below code, utilizing the `studres` and `hatvalues` commands from the `MASS` library, we can determine whether there exists leverage points and outlines.

```
## Outliers
stu.res <- studres(model.back.sel)
plot(stu.res, ylab = "Student Residuals", main = "Student Residuals vs. Index (Outliers)")
abline(3, 0, col="red")
abline(0, 0, col="blue")
abline(-3, 0, col="red")

## Leverage
hat.vals <- hatvalues(model.back.sel)
p <- length(survey) - 1
n <- nrow(survey)
plot(hat.vals, ylab = "Hat Values", main = "Hat Values vs. Index (Leverage)")
abline((p + 1)/n, 0, col="blue")
```



It's quite clear to see that the data set has multiple outliers and two particularly high leverage points. Outliers reside in the data set itself, being a point which is far from its predicted value. These points affect the model's adjusted R^2 value and RSE—adjusted R^2 decreases while RSE increases. Such an effect can more substantively be seen in the training process. It's possible that the outliers affect the RSE in such a way to skew the p-values for various predictors in the backwards selection phase, potentially changing the final model. Since outliers increase RSE, then a confidence interval for the intercept term, or any other coefficient, would be larger. The model's coefficients are likely are not changed drastically as a result of the outliers, since there are so few relative to the number of data points.

Leverage points, especially high leverage points, in great numbers, can invalidate a regression fit, and they can have a sizable impact on any of the model's coefficients. It's worth noting that there exists a data point that's both high leverage and an outlier: observation 595. This could be a particularly problematic observation!

```
outlier <- which(abs(stu.res) > 3)
leverage <- which(hat.vals > 0.03)
Reduce(intersect,list(outlier, leverage))
```

```
## [1] 595
```

7 Conclusion

Recall that the adjusted R^2 value of the selected model, the one obtained via backwards selection, is approximately 0.51. Although this value is at least 10% higher than the next closest model, such a value of 0.51 is not optimal by any means to make firm predictions—only half of the variability of the model can be explained! The previously mentioned outliers no doubt lower this value, so it's possible that this model could still make some meaningful predictions, should those data points be removed or otherwise ignored. Looking into the high leverage points might be wise as well.

This backwards selection model is merely multiple linear regression. It's possible that some predictors might produce better results if viewed in a polynomial fashion. For instance, the *Age vs. Log. Income* graph shows what appears to resemble a square root curve—one that increases but then rapidly slows.

The bottom line: in its current state, this model could make *some* predictions of quality but there will certainly be some poor predictions—recall how many total leverage points existed! However, given the nature of the data set, that being taken from a real life scenario, this model performs adequately enough, in my opinion, to give some insight into predicting income on a logarithmic scale.

Of the input variables, `hrs_work` and `edu` have a large impact on predicting the output variable. Generally speaking, the more hours an individual works, the higher they can expect their income to be. This suggestion corresponds to the correlation of `hrs_work` and `log_inc`—a positive one. We also saw that having higher education has an effect on one's income. We saw in an above graph that the higher one's education is, i.e., college or higher, they have a higher income. Again, this statement can be confirmed by looking at the correlation of `edu` and`edu`—a negative one—since a 1 for education actually trends with lower income individuals.