StatR 201: Winter 2013

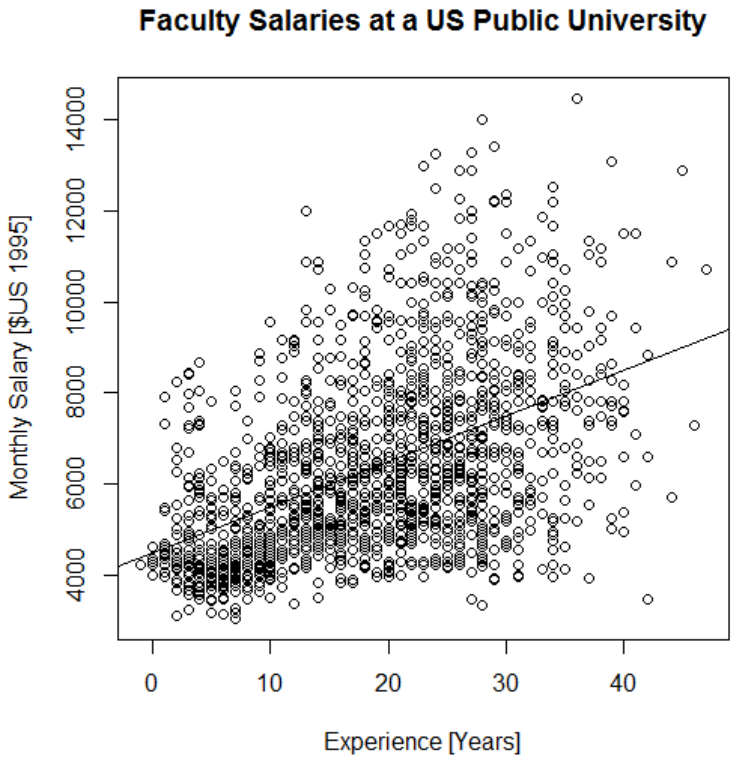
Final Project

Rod Doe

March 28, 2013

Overview

This project studies a dataset of faculty salaries at a public US university to search for salary gender bias. The data were collected on 1597 faculty members in 1995. The monthly salary is associated with these attributes – gender, highest degree attained, year of highest degree, field, year hired, and administrative duties. The calculated covariate "experience" is simply the result of (1995 – degree year). Here is a summary of the data:



Data Dictionary

Column	Description
id	surrogate key identity for the faculty member
yrdeg	1900 – year the highest degree was attained

startYr	1900 – the year that the faculty member was hired by the university
admin	Boolean indicator of whether faculty member has administrative duties
salary	Monthly salary in US dollars 1995
sex	1 = female, 2 = male
degree	highest degree attained 1=Bachelor's or Master's, 2 = PhD., 3 = Professional degree, i.e., medicine or law
field	1 = Arts and Humanities, 2 = Other, 3 = professional school (Business, Law, Engineering, Public Affairs)
rank	1 = Assistant, 2 = Associate, 3 = Full
experience	1995 – yrdeg
logsal	Natural log of salary column

Data Preparation, Exploration, and Sanity Check

This is a Stata dataset. Fortunately, the R library foreign enables consumption of Stata data by R.

```
setwd("C:/Users/Rod/SkyDrive/R/201/Project")
library(foreign)
salary = read.dta("salary-stata.dta")

str(salary)
table(salary$yrdeg)
table(salary$startyr)
table(salary$admin)
plot(1:length(salary$salary), sort(salary$salary))
table(salary$sex)
table(salary$degree)
table(salary$field)
table(salary$rank)
plot(1:length(salary$experience), sort(salary$experience))
which(abs(log(salary$salary) - salary$logsal) > 0.01)
```

The data appear to be in a good state with no singularities or data entry errors evident.

In the plot above, the data appear to exhibit heteroscedasticity. (I love that term. It has a freight train load of syllables, comes from the Greek terms "hetero" (different) and "skedasis" (dispersion), but probably shares those erudite roots with the decidedly un-lofty English word 'skedaddle', as in "Merle and me are gonna skedaddle down to Tacoma for the Monster Truck show."

Test for Heteroscedasticity

library(car)

ncvTest(lm(salary ~ experience, data=salary))

```
> ncvTest(lm(salary ~ experience, data=salary))
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 120.253    Df = 1     p = 5.568709e-28
```

This is sub-optimal. Perhaps the log of salary would exhibit less heteroscedasticity.

```
> ncvTest(lm(logsal ~ experience, data=salary))
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 31.21431    Df = 1     p = 2.310566e-08
```

We may have to live some dispersion.

Test for Variance Inflation Factors

Check for Variance Inflation Factors with a dataset that does not contain:

- Correlated columns
  - Columns salary and log(salary) correlate, so remove one.
  - Columns experience and yrdeg correlate, so remove one .
- Factors

```
> vif(salary.train.factorless)
     covariate    vif
[1,] "logsal"     "1.99707529566246"
[2,] "startYr"    "2.89509935201932"
[3,] "experience" "3.73415231458683"
[4,] "admin"      "1.10028110079599"
[5,] "degree"     "1.12505650407315"
[6,] "rank"       "2.70137280010313"
[7,] "field"      "1.03058831205255"
[8,] "gender"     "1.14628699917417""
```

With a threshold of 5, no VIF problems are evident in these data.

Analysis

Model Selection

In this section, sets of covariates will be cross-validated using various modeling techniques to determine which modeling technique is best. Techniques include:

- Linear models
  - Step forward
    - AIC
    - BIC
  - Step backward
    - AIC
    - BIC

- Generalized linear models
    - Step forward
        - AIC
        - BIC
    - Step backward
        - AIC
        - BIC
- Basic spline with three degrees of freedom.
    - Interesting note about the spline degrees of freedom – I set them to what seemed to make sense, but R increased them to 3. For some things, such as gender, three degrees of freedom seemed inherently unnecessary.

I initially included lasso but omitted it because the fit was simply dreadful.

What's Interesting?

One way to determine what's interesting in the data is to use the **regsubsets** function in the **leaps** library. (I don't think we covered this in class.) It prioritizes the covariates:

```
library(leaps)
```

```
> priorities <- regsubsets(logsal ~ ., data=salary.train.factorless)
> summary(priorities)
Subset selection object
Call: regsubsets.formula(logsal ~ ., data = salary.train.factorless)
7 Variables  (and intercept)
           Forced in Forced out
startYr        FALSE      FALSE
experience     FALSE      FALSE
admin          FALSE      FALSE
degree         FALSE      FALSE
rank           FALSE      FALSE
field          FALSE      FALSE
gender         FALSE      FALSE
1 subsets of each size up to 7
Selection Algorithm: exhaustive
         startYr experience admin degree rank field gender
1  ( 1 ) " "     " "        " "   " "    "*"  " "   " "
2  ( 1 ) " "     " "        "*"   " "    "*"  " "   " "
3  ( 1 ) " "     " "        "*"   " "    "*"  " "   "*"
4  ( 1 ) " "     " "        "*"   "*"    "*"  " "   "*"
5  ( 1 ) "*"     "*"        "*"   " "    "*"  " "   "*"
6  ( 1 ) "*"     "*"        "*"   "*"    "*"  " "   "*"
7  ( 1 ) "*"     "*"        "*"   "*"    "*"  "*"   "*"
```

From these results, I conclude that the order to add covariates to the model is:

1. rank
2. admin
3. gender
4. degree
5. (experience, startYr)
6. field

I also conclude that gender is important.

Cross Validation to Determine Optimum Model Method

I created a cross-validation function that iterates through a set of modeling techniques, and accumulates the RMS error of each technique in a dataset. In about a minute, it gives me the RMS error for many different modeling techniques. See attached code (crossValidate.*) for details. To successfully execute cross-validation, it was necessary to convert factors to numeric values.

For initial assessment, I created a dataset that contained the log of the salary as a function of these covariates: startYr, experience, admin, degree, rank, field, and gender, and their interactions. Using the result from above, I use this function to evaluate models with minimal covariates, and added covariates to determine the effect on the root mean square error.

Covariates rank, gender

```
formula.lm = as.formula("logsal ~ (rank + gender)^2")
formula.lm.bs = as.formula(logsal ~ bs(rank, df=3) + bs(gender, df=3))
results = crossValidate(salary.train.factorless, formula.lm, formula.lm.bs, nSets=10, "logsal")
> results
                      description rms.error
1                        model=lm 0.2230726
2                       model=glm 0.2230726
3     model=lm, step=forward, AIC 0.2230726
4    model=lm, step=backward, AIC 0.2230726
5     model=lm, step=forward, BIC 0.2230726
6    model=lm, step=backward, BIC 0.2233178
7    model=glm, step=forward, AIC 0.2230726
8   model=glm, step=backward, AIC 0.2230726
9    model=glm, step=forward, BIC 0.2230726
10 model=glm, step=backward, BIC 0.2233178
11                    model=lm.bs 0.2195397
```

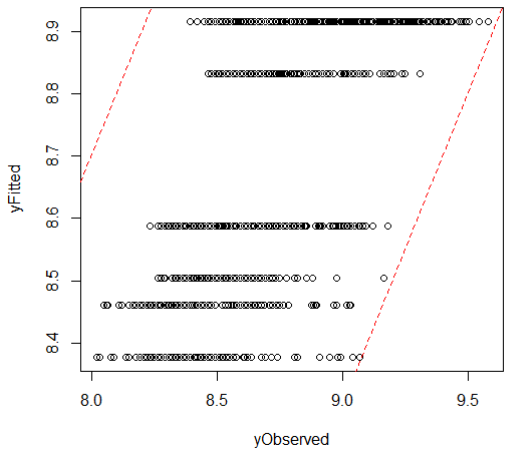


Figure 1: logsal ~ bs(rank, df=3) + bs(gender, df=3)

Covariates rank, admin, gender

```
formula.lm = as.formula("logsal ~ (rank + admin + gender)^2")
formula.lm.bs = as.formula(logsal ~ bs(rank, df=3) + bs(admin, df=3) + bs(gender, df=3))
results = crossValidate(salary.train.factorless, formula.lm, formula.lm.bs, nSets=10, "logsal")
> results
                      description rms.error
1                        model=lm 0.2175399
2                       model=glm 0.2175399
3     model=lm, step=forward, AIC 0.2175399
4    model=lm, step=backward, AIC 0.2177437
5     model=lm, step=forward, BIC 0.2175399
```

```
6    model=lm, step=backward, BIC 0.2175649
7    model=glm, step=forward, AIC 0.2175399
8   model=glm, step=backward, AIC 0.2177437
9    model=glm, step=forward, BIC 0.2175399
10 model=glm, step=backward, BIC 0.2175649
11                   model=lm.bs 0.2137611
```

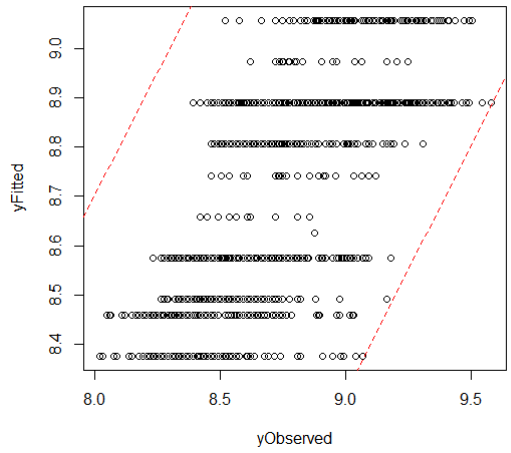


Figure 2: logsal ~ bs(rank, df=3) + bs(admin, df=3) + bs(gender, df=3)

Covariates rank, admin, gender, degree

formula.lm = as.formula("logsal ~ (rank + admin + gender + degree)^2")

formula.lm.bs = as.formula(logsal ~ bs(rank, df=3) + bs(admin, df=3) + bs(gender, df=3) + bs(degree, df=3))

results = crossValidate(salary.train.factorless, formula.lm, formula.lm.bs, nSets=10, "logsal")

```
> results
                    description rms.error
1                      model=lm 0.2188112
2                     model=glm 0.2188112
3    model=lm, step=forward, AIC 0.2188112
4   model=lm, step=backward, AIC 0.2190007
5    model=lm, step=forward, BIC 0.2188112
6   model=lm, step=backward, BIC 0.2186530
7   model=glm, step=forward, AIC 0.2188112
8  model=glm, step=backward, AIC 0.2190007
9   model=glm, step=forward, BIC 0.2188112
10 model=glm, step=backward, BIC 0.2186530
11                   model=lm.bs 0.2127437
```

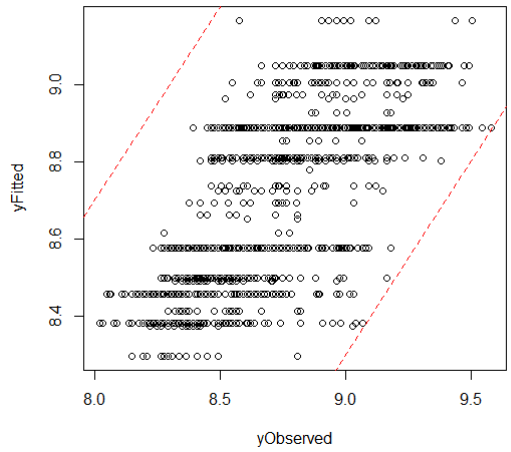


Figure 3: logsal ~ bs(rank, df=3) + bs(admin, df=3) + bs(gender, df=3) + bs(degree, df=3)

Covariates rank, admin, gender, degree, experience

formula.lm = as.formula("logsal ~ (rank + admin + gender + degree + experience)^2")

formula.lm.bs = as.formula(logsal ~ bs(rank, df=3) + bs(admin, df=3) + bs(gender, df=3) + bs(degree, df=3) + bs(experience, df=3))

results = crossValidate(salary.train.factorless, formula.lm, formula.lm.bs, nSets=10, "logsal")

```
> results
                     description rms.error
1                       model=lm 0.2152631
2                      model=glm 0.2152631
3    model=lm, step=forward, AIC 0.2152631
4   model=lm, step=backward, AIC 0.2145457
5    model=lm, step=forward, BIC 0.2152631
6   model=lm, step=backward, BIC 0.2149378
7   model=glm, step=forward, AIC 0.2152631
8  model=glm, step=backward, AIC 0.2145457
9   model=glm, step=forward, BIC 0.2152631
10 model=glm, step=backward, BIC 0.2149378
11                   model=lm.bs 0.2114908
```

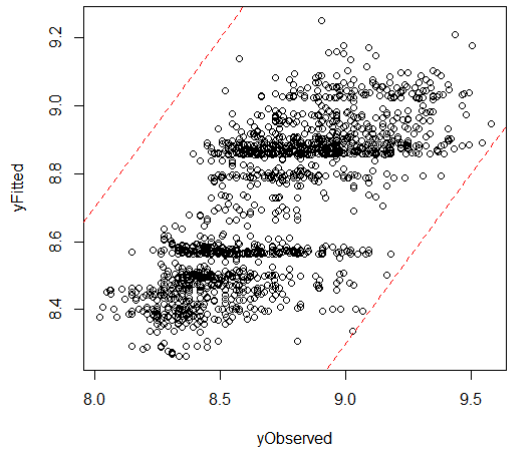


Figure 4: logsal ~ bs(rank, df=3) + bs(admin, df=3) + bs(gender, df=3) + bs(degree, df=3) + bs(experience, df=3)

Covariates rank, admin, gender, degree, experience, startYr

formula.lm = as.formula("logsal ~ (rank + admin + gender + degree + experience + startYr)^2")

formula.lm.bs = as.formula(logsal ~ bs(rank, df=3) + bs(admin, df=3) + bs(gender, df=3) + bs(degree, df=3) + bs(experience, df=3) + bs(startYr, df=3))

results = crossValidate(salary.train.factorless, formula.lm, formula.lm.bs, nSets=10, "logsal")

```
> results
                    description rms.error
1                      model=lm 0.2129284
2                     model=glm 0.2129284
3    model=lm, step=forward, AIC 0.2129284
4   model=lm, step=backward, AIC 0.2128521
5    model=lm, step=forward, BIC 0.2129284
6   model=lm, step=backward, BIC 0.2127679
7   model=glm, step=forward, AIC 0.2129284
8  model=glm, step=backward, AIC 0.2128521
9   model=glm, step=forward, BIC 0.2129284
10 model=glm, step=backward, BIC 0.2127679
11                  model=lm.bs 0.2087033
```

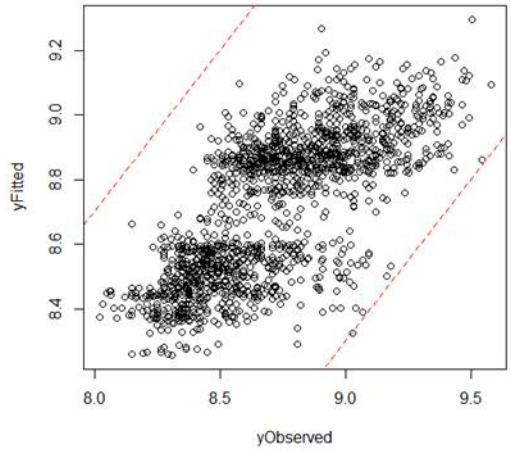


Figure 5: logsal ~ bs(rank, df=3) + bs(admin, df=3) + bs(gender, df=3) + bs(degree, df=3) + bs(experience, df=3) + bs(startYr, df=3)

Covariates rank, admin, gender, degree, experience, startYr, field

formula.lm = as.formula("logsal ~ (rank + admin + gender + degree + experience + startYr + field)^2")

formula.lm.bs = as.formula(logsal ~ bs(rank, df=3) + bs(admin, df=3) + bs(gender, df=3) + bs(degree, df=3) + bs(experience, df=3) + bs(startYr, df=3) + bs(field, df=3))

results = crossValidate(salary.train.factorless, formula.lm, formula.lm.bs, nSets=10, "logsal")

```
> results
                     description rms.error
1                       model=lm 0.2118849
2                      model=glm 0.2118849
3    model=lm, step=forward, AIC 0.2118849
4   model=lm, step=backward, AIC 0.2125965
5    model=lm, step=forward, BIC 0.2118849
6   model=lm, step=backward, BIC 0.2127815
7   model=glm, step=forward, AIC 0.2118849
8  model=glm, step=backward, AIC 0.2125965
9   model=glm, step=forward, BIC 0.2118849
10 model=glm, step=backward, BIC 0.2127815
11                   model=lm.bs 0.1962729
```

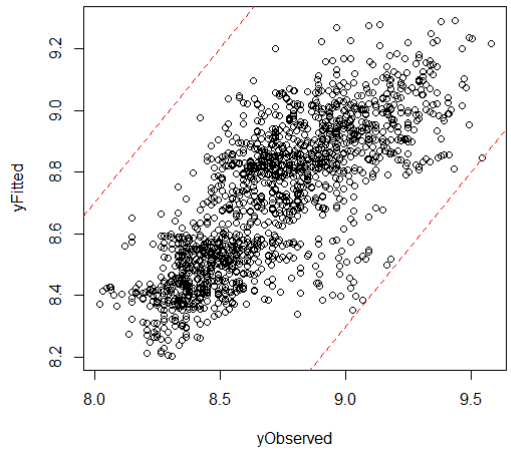


Figure 6: logsal ~ bs(rank, df=3) + bs(admin, df=3) + bs(gender, df=3) + bs(degree, df=3) + bs(experience, df=3) + bs(startYr, df=3) + bs(field, df=3)

ANOVA shows that all these variables are significant predictors of logsal:

```
> formula.lm.bs = as.formula(logsal ~ bs(rank, df=3) + bs(admin, df=3) +
bs(gender, df=3) + bs(degree, df=3 ) + bs(experience, df=3) + bs(startYr,
df=3) + bs(field, df=3) )
> model = lm(formula = formula.lm.bs, data=salary.train.factorless)
> anova(model)
Analysis of Variance Table

Response: logsal
                        Df Sum Sq Mean Sq F value    Pr(>F)
bs(rank, df = 3)         2 67.523  33.762 903.249 < 2.2e-16 ***
bs(admin, df = 3)        1  3.983   3.983 106.558 < 2.2e-16 ***
bs(gender, df = 3)       1  1.982   1.982  53.017 5.189e-13 ***
bs(degree, df = 3)       2  2.410   1.205  32.241 1.887e-14 ***
bs(experience, df = 3)   3  1.127   0.376  10.047 1.473e-06 ***
bs(startYr, df = 3)      3  2.221   0.740  19.804 1.329e-12 ***
bs(field, df = 3)        2  8.140   4.070 108.887 < 2.2e-16 ***
Residuals             1582 59.132   0.037
```

Since all these variables are significant predictors, I use them all in the prediction model.

Prediction

I broke the salary data set into a training and test set, with a random sample of 75% of the data being dedicated to training. The remainder was reserved for the test set.

Prediction for Model Evaluation

The model was created using training data:

```
formula.lm.bs = as.formula(logsal ~ bs(rank, df=3) + bs(admin, df=3) +
bs(gender, df=3) + bs(degree, df=3 ) + bs(experience, df=3) + bs(startYr,
df=3) + bs(field, df=3))

model.lm.bs = lm(formula.lm.bs, data=salary.train.factorless)
```

I then fit the test data to the training model, and superimposed the observed test data and fitted training data. The expectation is that the two sets of data should "play well together", meaning that they should be indistinguishable. See Figure 7: Qualitative Depiction of Test Data Fit to Training Data Model below.

I also plotted the observed test data against fitted test data. The data are bounded by lines with slope of 1 and intercepts of (1-,1).

```
plot(salary.test.factorless$experience, predict(model.lm.bs,
salary.test.factorless), col=rgb(0,1,0),
      xlab="Experience [Years]", ylab="log(salary) [$US 1995]",
main="Observed (Train) and Fitted (Test) Data")
points(salary.train.factorless$experience, salary.train.factorless$logsal,
col=rgb(0,0,1))
legend("topleft", legend=c("Test", "Train"), col=c(rgb(0,1,0), rgb(0,0,1)),
pch=19)
```

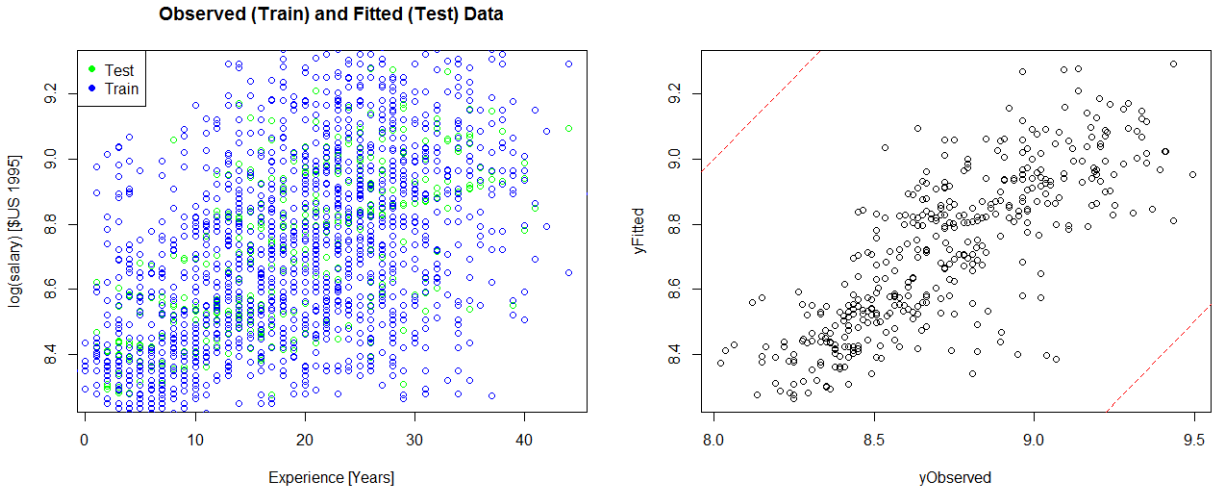


Figure 7: Qualitative Depiction of Test Data Fit to Training Data Model

The observed training data used to construct the model and the test data fitted with that model are nicely comingled. Absent color coding, the two sets would be indistinguishable.

Evidence of Gender Bias – A Simple Experiment

I conducted an experiment to expose evidence of gender bias with these simple steps:

- Create a model of the salary that does not include gender.
- Compare the observed salary with those fitted by a genderless model. Specifically:
  - Get the count of male and female salaries that increased with gender removed from the model.
  - Get the count of male and female salaries that decreased with gender removed from the model.

Model	Female Increase	Female Decreased	Male Increased	Male Decreased
Genderless	261	148	581	607

The pink concentration above 0, and the blue concentration below 0 depict the effect.

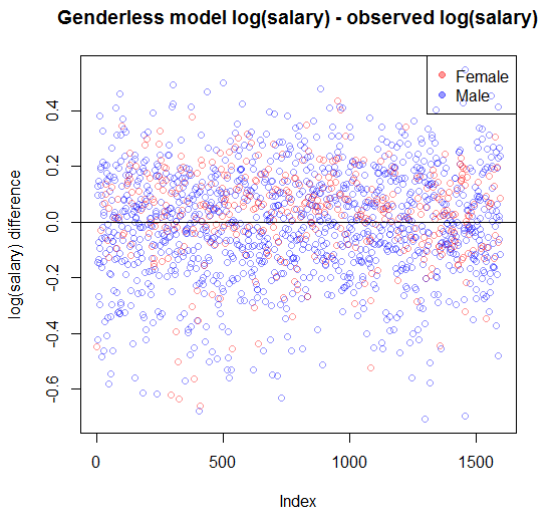


Figure 8: Who gets a raise when salary is modeled without gender?

The simple experiment indicates that when gender is removed from the model, more female salaries increased than decreased, whereas more male salaries decreased than increased. The code for this is attached. The fact that a significant fraction of the female salaries decreased when gender was removed from the model indicates that this is a complex issue. A simple N% raise for all females would not fix this problem.

Prediction for Detection of Gender Bias

For prediction, I created gender-specific datasets that I used as the "newdata" parameter for the predict function. I predicted log(salary) given hypothetical values of the predictors. This allowed me to create data sets that answered questions such as "What would be the starting salary of a female Ph.D. with 15 years of experience in her field?" This sample code demonstrates the generation of data for male new hires.

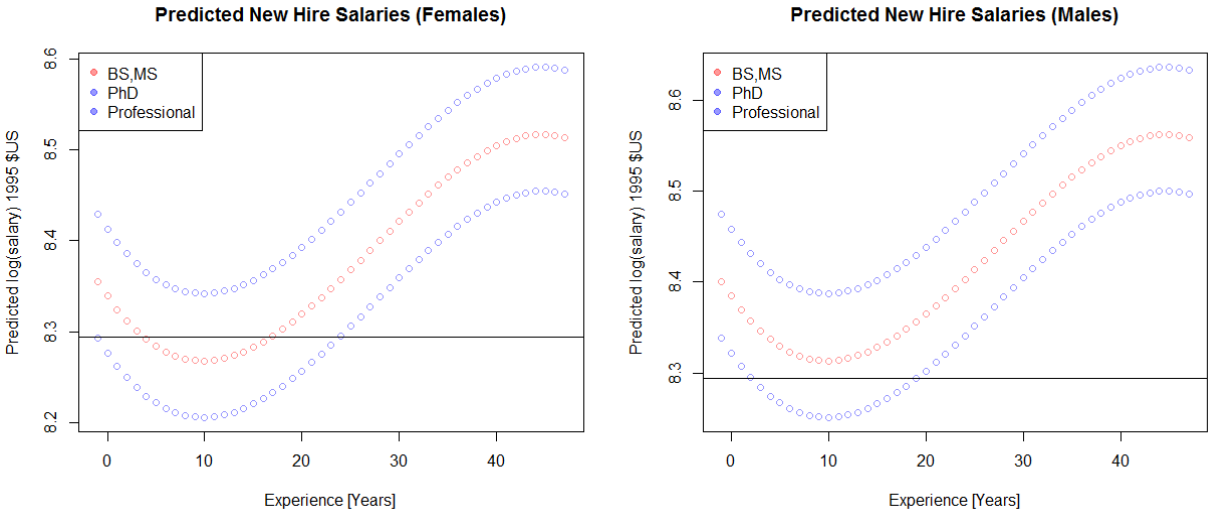
```
# Create a dataframe of male new hires of all degrees.
male.new.hires <- data.frame(startYr = numeric(), experience=numeric(),
admin=numeric(), degree=numeric(), rank=numeric(), field=numeric(),
gender=numeric())

for (degree in range.degree[1]:range.degree[2]) {
      for (experience in range.experience[1]:range.experience[2]) {
            new.row <- data.frame(  startYr=range.startYr[2],
                                    experience=experience,
                                    admin=range.admin[1],
                                    degree=degree,
                                    rank=range.rank[1],
                                    field=range.field[1],
                                    gender=range.gender[2] )
            male.new.hires <- rbind(male.new.hires, new.row)
      }
}

plot(male.new.hires$experience, predict(model.lm.bs, male.new.hires),
            xlab="Experience [Years]",
            ylab="Predicted log(salary) 1995 $US",
            main="Predicted New Hire Salaries (Males)",
            col=cols[male.new.hires$degree])
```

Predicted Salary for New Hires in 1995

For new hires, there appears to be clear evidence of a gender bias.



The horizontal line is through the log(4000), and appears for comparison purposes. The key point to note is that $4000/month appears lower on the chart for males. I am perplexed by the shape of this curve. The decrease for in log(salary) for experience between 0 and 10 seems counterintuitive. I suspect some spline skullduggery is the root cause.

Live by the sword, die by the sword. And the spline as well, perhaps.

Summary

As a result of this project, I built a monster cross-validation function that I suspect I will use again, like maybe next week. I will incorporate it into a personal library for future use.

The data show a general gender bias in favor of male faculty. It is a complex problem that will not be fixed by giving all females a raise. Further investigation is warranted in order to fully understand the source/application of the bias.

I have completed UW certificates in Software Engineering, XML Standards and Technologies, Project Mangement, C# & .NET Development, and SQL Server Specialist. This certificate, and this class in particular, has been by far the most difficult. Thank you for the challenge.