Assignment 4

ECON 308: Econometrics

Elizabeth Goodwin

11/21/2021

Question 1

What is the justification for using LOOCV to estimate the predictive accuracy of a model?

LOOCV is justified to estimate the predictive accuracy of a model because it manages to do cross-validation on a nearly full dataset without losing a significant amount of the dataset. In other forms of cross validation, such as splitting into a testing/training set or using K-fold validation, you nearly always lose a very large amount of your dataset for training in the first place. Leave one out uses all of a dataset but one item, and rotates that left out item to each item in the dataset. While this is very resource intensive, it is also very accurate.

Question 2

Under what circumstances is K-fold cross-validation preferable to LOOCV for model selection? When might AIC and BIC be preferable to either

K-fold validation, while less accurate than LOOCV for model selection, has one key advantage: resource usage. LOOCV is incredibly resource intensive, and that intensity can get unreasonably high with larger datasets or more resource intensive models. While LOOCV works with smaller datasets and simple linear models, the more intensive the model the more time it will take up. K-Fold validation allows you to capture a lot of the advantages of LOOCV (using most of your dataset for cross validation) but with far less resource intensity. You pay for this by less accurate model selection, but in many applications the benefits far outweigh the costs.

AIC or BIC also have some advantages compared to both of these selection mechanisms. First off, they use very little computational resources. While with our current technology this is less of an issue, it can still be an issue for very large datasets. K-fold uses less resources compared to LOOCV, but AIC/BIC use nearly no resources. Running K-Fold validation on large datasets can still take a long time, and AIC/BIC can be an easy way to compare models.

Question 3

Load the dataset, drop 0 observations for education and marriage, and create encoded categorical variables.

```
use "credit_default.dta"

drop if education == 0
drop if marriage == 0

gen sex_enc = 0
replace sex_enc = 1 if sex == 2


gen has_ba = 0
replace has_ba = 1 if education <= 2

gen married = 0
replace married = 1 if marriage == 1
```

Next I generated values showing the fraction repaid every month, and then replaced missing values with zero. I also generated values showing the percent of the credit limit for each bill, and generated a value averaging the 6 variables.

```
gen tot_bill = 0
gen tot_pay = 0
foreach n of numlist 1/6 {
  gen payshare`n' = pay_amt`n' / bill_amt`n'
  replace payshare`n' = 0 if(payshare`n' == .)
    replace tot_bill = tot_bill + bill_amt`n'
    replace tot_pay = tot_pay + pay_amt`n'
}
gen frac_paid = tot_pay / tot_bill

gen tot_bill_CL_share = 0
foreach n of numlist 1/6 {
    gen bill_CL_share`n' = bill_amt`n' / limit_bal
    replace tot_bill_CL_share = tot_bill_CL_share + bill_CL_share`n'
}
gen avg_bill_CL_share = tot_bill_CL_share / 6
```

I also changed the pay_n variables to factor variables, as the negative values don't work as factor variables in stata regressions.

```
    foreach n of numlist 0 2 3 4 5 6 {
        tostring pay_`n', replace
        egen payc_`n' = group(pay_`n')
    }
```

Models

Model 1

This is the first model I used.

```
reg defaultpaymentnextmonth limit_bal i.sex i.education i.marriage age frac_paid  ///
i.payc_0 i.payc_2 i.payc_3 i.payc_4 i.payc_5 i.payc_6 payshare1 payshare2  ///
payshare3 payshare4 payshare5 payshare6 i.education#c.limit_bal ///
i.marriage#c.limit_bal c.age#c.limit_bal c.age#c.frac_paid
```

This model primarily focuses on the status of payment for each month, and the share of the amount owed. It also adds some interaction terms.

Model 2

This model is similar to the model above, but with a lot more variables. I included all of the bill share of the credit limit as well.

```
reg defaultpaymentnextmonth limit_bal i.sex i.education i.marriage age frac_paid i.payc_0 ///
i.payc_2 i.payc_3 i.payc_4 i.payc_5 i.payc_6 payshare1 payshare2 payshare3 payshare4 ///
payshare5 payshare6 i.education#c.limit_bal i.marriage#c.limit_bal c.age#c.limit_bal ///
c.age#c.frac_paid bill_CL_share1 bill_CL_share2 bill_CL_share3 bill_CL_share4  ///
bill_CL_share5 bill_CL_share6 avg_bill_CL_share c.avg_bill_CL_share#c.age  ///
c.avg_bill_CL_share#i.education
```

Results

	(1) Model 1		(2) Model 2		(3) LASSO
LIMIT_BAL	-0.000000411**	(0.000000135)	-0.000000437**	(0.000000135)	-0.000000538
SEX=2	-0.0220***	(0.00631)	-0.0229***	(0.00631)	
EDUCATION=2	-0.0165	(0.0163)	-0.0162	(0.0163)	
EDUCATION=3	-0.00138	(0.0223)	-0.000540	(0.0223)	-0.00305
EDUCATION=4	-0.0689	(0.139)	-0.0663	(0.138)	-0.0377
EDUCATION=5	-0.189*	(0.0811)	-0.193*	(0.0808)	-0.0957
EDUCATION=6	0.0157	(0.197)	0.0134	(0.197)	
MARRIAGE=2	-0.0306**	(0.0115)	-0.0317**	(0.0115)	-0.0259
MARRIAGE=3	-0.0235	(0.0467)	-0.0148	(0.0475)	
AGE	-0.000451	(0.000842)	-0.000471	(0.000842)	0.000568
frac_paid	-0.00392	(0.00395)	-0.00370	(0.00394)	-0.000106
group(pay_0)=2	-0.0283	(0.0236)	-0.0267	(0.0234)	-0.0278
group(pay_0)=3	-0.110***	(0.0147)	-0.106***	(0.0147)	-0.0891
group(pay_0)=4	0.0709***	(0.0155)	0.0735***	(0.0154)	0.0585
group(pay_0)=5	0.344***	(0.0161)	0.347***	(0.0160)	0.351
group(pay_0)=6	0.364***	(0.0354)	0.365***	(0.0357)	0.346
group(pay_0)=7	0.311***	(0.0677)	0.331***	(0.0670)	0.261
group(pay_0)=8	0.162	(0.121)	0.167	(0.121)	0.164
group(pay_0)=9	-0.310	(0.246)	-0.301	(0.246)	
group(pay_0)=10	-0.0181	(0.408)	-0.0118	(0.408)	0.205
group(pay_0)=11	-0.122	(0.381)	-0.118	(0.381)	
group(pay_2)=2	-0.0247	(0.0239)	-0.0270	(0.0237)	-0.0452
group(pay_2)=3	0.0589***	(0.0170)	0.0584***	(0.0171)	
group(pay_2)=4	-0.0323	(0.123)	0.00854	(0.130)	-0.103
group(pay_2)=5	0.0712***	(0.0171)	0.0735***	(0.0171)	0.0359
group(pay_2)=6	0.0630	(0.0364)	0.0502	(0.0365)	0.0491
group(pay_2)=7	-0.0402	(0.0728)	-0.0349	(0.0731)	-0.0812
group(pay_2)=8	0.520*	(0.235)	0.509*	(0.234)	0.228
group(pay_2)=9	1.058*	(0.418)	1.057*	(0.418)	0.267
group(pay_2)=10	1.917**	(0.607)	1.948**	(0.601)	0.0466
group(pay_2)=11	-0.593	(0.427)	-0.668	(0.422)	
group(pay_3)=2	-0.0172	(0.0205)	-0.0173	(0.0206)	-0.00839
group(pay_3)=3	0.0119	(0.0154)	0.0111	(0.0154)	

group(pay_3)=4	0.285	(0.283)	0.240	(0.286)	-0.0304
group(pay_3)=5	0.0869***	(0.0171)	0.0875***	(0.0171)	0.0586
group(pay_3)=6	0.115*	(0.0464)	0.111*	(0.0466)	0.0686
group(pay_3)=7	-0.00810	(0.110)	-0.0242	(0.109)	-0.0219
group(pay_3)=8	-0.443*	(0.183)	-0.446*	(0.183)	-0.134
group(pay_3)=9	-0.938*	(0.402)	-0.950*	(0.398)	0.316
group(pay_3)=10	-0.0466	(0.147)	0.0185	(0.143)	0.0271
group(pay_3)=11	-0.779	(0.818)	-0.707	(0.797)	-0.0990
group(pay_4)=2	0.0399	(0.0213)	0.0426*	(0.0213)	
group(pay_4)=3	-0.00847	(0.0144)	-0.0151	(0.0143)	
group(pay_4)=4	0	(.)	0	(.)	0.273
group(pay_4)=5	0.0427*	(0.0184)	0.0342	(0.0184)	0.0525
group(pay_4)=6	0.0561	(0.0513)	0.0731	(0.0511)	0.0118
group(pay_4)=7	-0.126	(0.0937)	-0.0977	(0.0938)	0.0754
group(pay_4)=8	-0.606**	(0.210)	-0.607**	(0.205)	-0.182
group(pay_4)=9	0	(.)	0	(.)	-0.486
group(pay_4)=10	0.233	(0.287)	0.180	(0.264)	
group(pay_5)=2	-0.0111	(0.0213)	-0.0110	(0.0213)	
group(pay_5)=3	0.0236	(0.0138)	0.0327*	(0.0138)	
group(pay_5)=4	0.0799***	(0.0198)	0.0883***	(0.0197)	0.0572
group(pay_5)=5	0.142**	(0.0506)	0.118*	(0.0518)	0.0187
group(pay_5)=6	0.155	(0.115)	0.153	(0.110)	-0.0149
group(pay_5)=7	0.283	(0.209)	0.298	(0.191)	0.118
group(pay_5)=8	1.189*	(0.579)	1.170*	(0.578)	0.140
group(pay_5)=9	0	(.)	0	(.)	0.124
group(pay_6)=2	0.0168	(0.0163)	0.0125	(0.0162)	0.00216
group(pay_6)=3	-0.0536***	(0.0124)	-0.0593***	(0.0124)	-0.0252
group(pay_6)=4	0.0139	(0.0174)	0.00698	(0.0173)	0.0381
group(pay_6)=5	0.158***	(0.0455)	0.147**	(0.0462)	0.130
group(pay_6)=6	0.0625	(0.115)	0.0538	(0.111)	0.0153
group(pay_6)=7	-0.306	(0.234)	-0.310	(0.230)	
group(pay_6)=8	0.133	(0.209)	0.158	(0.187)	0.139
group(pay_6)=9	0.0388	(0.287)	0.0158	(0.260)	
payshare1	-0.0000202*	(0.00000940)	-0.0000237*	(0.00000964)	-0.0000209
payshare2	-0.00000977	(0.0000260)	-0.00000986	(0.0000260)	0.00000389
payshare3	-0.00000360	(0.00000489)	-0.00000359	(0.00000489)	0.000000159
payshare4	0.00000920	(0.0000164)	0.00000997	(0.0000162)	0.00000930
payshare5	0.0000204	(0.0000384)	0.0000249	(0.0000375)	
payshare6	-0.0000257	(0.0000705)	-0.0000275	(0.0000704)	0.00000717
EDU=2 × LIMIT_BAL	4.30e-08	(5.61e-08)	4.37e-08	(5.62e-08)	
EDU=3 × LIMIT_BAL	-8.29e-08	(8.30e-08)	-7.99e-08	(8.30e-08)	-9.25e-09
EDU=4 × LIMIT_BAL	-0.000000148	(0.000000478)	-0.000000152	(0.000000471)	-0.000000133
EDU=5 × LIMIT_BAL	0.000000380	(0.000000279)	0.000000419	(0.000000280)	4.90e-09
EDU=6 × LIMIT_BAL	-0.000000335	(0.000000661)	-0.000000345	(0.000000661)	
MAR=2 × LIMIT_BAL	4.55e-08	(5.31e-08)	5.65e-08	(5.31e-08)	
MAR=3 × LIMIT_BAL	0.000000310	(0.000000405)	0.000000247	(0.000000408)	
AGE × LIMIT_BAL	4.37e-09	(3.09e-09)	5.09e-09	(3.10e-09)	7.97e-10
AGE × frac_paid	0.0000933	(0.000128)	0.0000864	(0.000128)	
bill_CL_share1	0.0162	(0.0226)	0.0137	(0.0227)	
bill_CL_share2	0.0121	(0.0268)	0.0209	(0.0267)	0.00408
bill_CL_share3	0.0203	(0.0251)	0.0188	(0.0250)	0.00667
bill_CL_share4	-0.0289	(0.0265)	-0.0324	(0.0265)	
bill_CL_share5	0.00494	(0.0286)	0.00598	(0.0285)	-0.0109
bill_CL_share6	0.00126	(0.0234)	0.00884	(0.0231)	

avgbillCLshare	0	(.)	0	(.)	
avgbillCLshare × AGE	-0.000390	(0.00103)	-0.000660	(0.00103)	
EDU=2 × avgbillCLshare	0.0349	(0.0217)	0.0340	(0.0217)	0.0101
EDU=3 × avgbillCLshare	0.0228	(0.0294)	0.0197	(0.0296)	
EDU=4 × avgbillCLshare	0.00317	(0.179)	0.00658	(0.181)	
EDU=5 × avgbillCLshare	0.0324	(0.0959)	0.00997	(0.0957)	
EDU=6 × avgbillCLshare	-0.0821	(0.188)	-0.109	(0.190)	-0.0470
SEX=1					0.0161
MARRIAGE=1					0.00166
group(pay_2)=1					-0.0294
group(pay_3)=1					-0.00910
group(pay_4)=1					-0.00451
group(pay_4)=11					-0.608
group(pay_5)=1					-0.00920
group(pay_6)=10					0.945
MAR=1 × LIMIT_BAL					-2.92e-08
limit_bal_2					5.76e-13
bill_amt1_2					7.76e-13
bill_amt2_2					8.78e-13
bill_amt3_2					1.25e-13
bill_amt4_2					6.63e-14
bill_amt5_2					9.21e-13
bill_amt5_3					-1.66e-18
bill_amt6_2					7.88e-14
bill_amt6_3					-4.07e-19
pay_amt1_2					6.52e-13
pay_amt3_2					1.05e-12
pay_amt4_2					5.03e-13
pay_amt5_2					1.73e-12
pay_amt6_2					4.78e-13
tot_bill_3					-4.23e-21
frac_paid_3					-1.56e-09
limt_bal_bill_amt2					-5.23e-13
limt_bal_bill_amt3					-3.60e-14
limt_bal_pay_amt1					-1.31e-13
limt_bal_pay_amt2					3.00e-13
limt_bal_pay_amt4					4.11e-13
limt_bal_pay_amt6					3.24e-13
limt_bal_frac_paid					1.77e-09
age_bill_amt1					-3.19e-09
age_pay_amt1					-5.48e-09
age_pay_amt2					-3.39e-10
age_pay_amt4					-8.21e-10
age_pay_amt5					-3.19e-09
age_tot_pay					-9.78e-09
Constant	0.249***	(0.0351)	0.250***	(0.0351)	0.276
Observations	14248		14254		29064
Adjusted R^2	0.226		0.225		

Standard errors in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 2: Model Validation

Model	AIC	BIC	K-fold(k=40, in MSE)
Model 1	23859.08	23859.08	.13340243
Model 2	23864.35	24658.97	.1333883

From this is appears that while both models are quite similar overall, there are some differences in model validation. AIC has model 1 being slightly better, with a small increase. BIC has model 1 being better by a larger amount. K-fold validation, however, is extremely similar and has model 2 being ever so slightly better, although it is small enough to not really be significant. The models might just be too similar to really make much of a difference either way.

Testing on half of dataset

Table 3: Model Validation on half of dataset

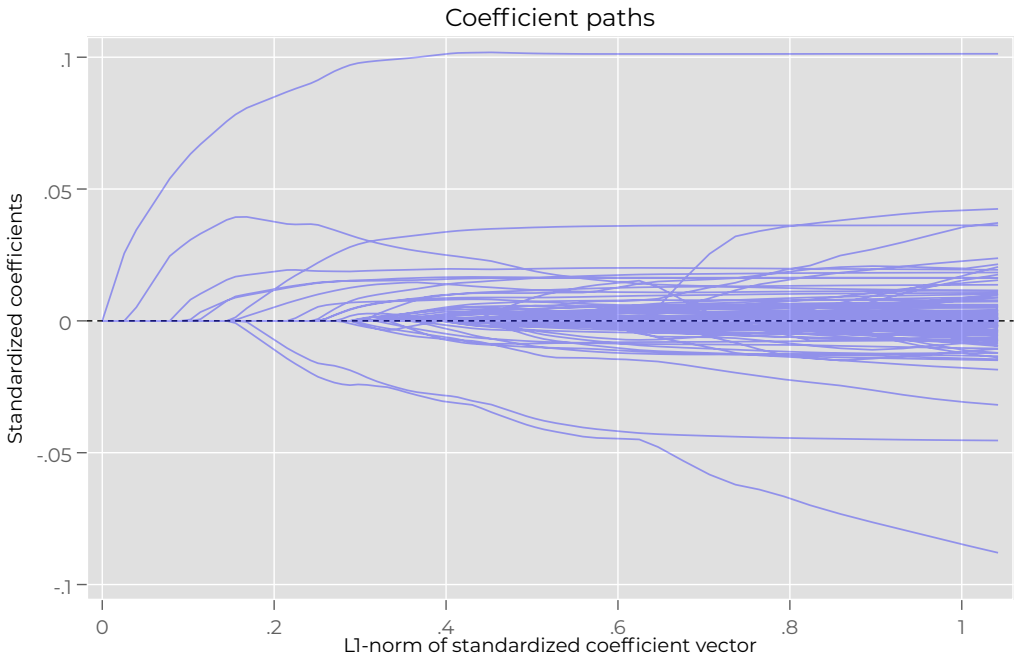
Model	MSE
Model 1 (In sample)	.1339462
Model 1 (Out of sample)	.1318714
Model 2 (In sample)	.1337864
Model 2 (Out of sample)	.1319985

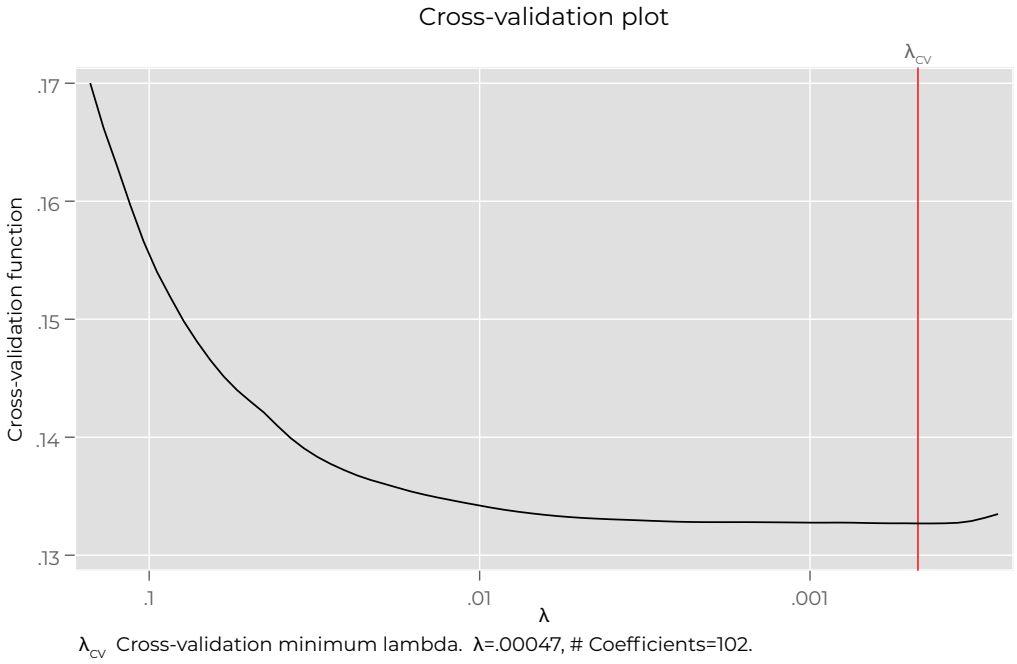
Model 2 worked better on the in the in sample dataset, while model 1 worked better in the out of sample dataset. This difference is relatively small, however. Model 1 is a simpler dataset, so the result makes sense.

Lasso

I used this very rich model to feed a lasso with 40 k-folds. The coefficients exist on the coefficient table on the previous page, as the third(and longest) entry. It has a mean squared error of `.1315139`. It used 102 of the available 168, dropping the rest.

```
lasso linear defaultpaymentnextmonth limit_bal i.sex i.education i.marriage age frac_paid \\\
i.payc_0 i.payc_2 i.payc_3 i.payc_4 i.payc_5 i.payc_6 payshare1 payshare2 payshare3 \\\
payshare4 payshare5 payshare6 i.education#c.limit_bal i.marriage#c.limit_bal  \\\
c.age#c.limit_bal c.age#c.frac_paid bill_CL_share1 bill_CL_share2 bill_CL_share3 \\\
bill_CL_share4 bill_CL_share5 bill_CL_share6 avg_bill_CL_share c.avg_bill_CL_share#c.age \\\
c.avg_bill_CL_share#i.education limit_bal_2 limit_bal_3 bill_amt1_2 bill_amt1_3 \\\
bill_amt2_2 bill_amt2_3 bill_amt3_2 bill_amt3_3 bill_amt4_2 bill_amt4_3 bill_amt5_2 \\\
bill_amt5_3 bill_amt6_2 bill_amt6_3 pay_amt1_2 pay_amt1_3 pay_amt2_2 pay_amt2_3  \\\
pay_amt3_2 pay_amt3_3 pay_amt4_2 pay_amt4_3 pay_amt5_2 pay_amt5_3 pay_amt6_2  \\\
pay_amt6_3 tot_bill_2 tot_bill_3 tot_pay_2 tot_pay_3 frac_paid_2 frac_paid_3  \\\
limt_bal_age limt_bal_bill_amt1 limt_bal_bill_amt2 limt_bal_bill_amt3 \\\
limt_bal_bill_amt4 limt_bal_bill_amt5 limt_bal_bill_amt6 limt_bal_pay_amt1 \\\
limt_bal_pay_amt2 limt_bal_pay_amt3 limt_bal_pay_amt4 limt_bal_pay_amt5  \\\
limt_bal_pay_amt6 limt_bal_tot_bill limt_bal_tot_pay limt_bal_frac_paid age_bill_amt1 \\\
age_bill_amt2 age_bill_amt3 age_bill_amt4 age_bill_amt5 age_bill_amt6 age_pay_amt1 \\\
age_pay_amt2 age_pay_amt3 age_pay_amt4 age_pay_amt5 age_pay_amt6 age_tot_bill  \\\
age_tot_pay age_frac_paid, folds(40)
```





λ_{CV} Cross-validation minimum lambda. λ=.00047, # Coefficients=102.

The most important variables mostly came from the pay variables. pay_0 equaling 3, or payment delay for three months, had one of the largest impacts, with pay_0=2 following closely behind. Larger gaps of payments also had large impacts. Pay values for other months were important as well, with pay3=5 being very important. The largest coefficient of all came from pay_6=7, with a .945 coefficient. this is interesting, as it shows a very long payment delay from a long time ago. So it makes sense that it would be large. Biggest changes that lasso did was getting rid of some of the very large coefficients in the pay_3 variables. One went from -.95 to .316, for instance. another went from .24 to -.03. Honestly though, the MSE of the lasso wasn't all that different from the MSE of the other models. While it was better, it wasn't really a huge impact.

Question 4

Table 4: Hypothetical data on potential outcomes and treatment status

Student	Y_1	Y_0	δ	D
1	8	9	-1	0
2	9	5	4	0
3	5	6	-1	1
4	8	5	3	1
5	7	2	5	1
6	1	1	0	0

(a) Unit treatment effect is simply the outcome if they were treated (Y_1) minus the outcome if they were not treated (Y_0) for each individual. I filled out the results in column δ above.

(b) Average Treatment Effect (ATE) is simply the average of all the unit treatment effect, or:

$$\frac{1}{N}\sum_{i=1}^{N}(Y_1 i - Y_0 i)$$

For these outcomes, the average treatment effect is $1.\overline{666}$

```
ATE <- mean(table$Unit_treatment)
ATE
```

```
[1] 1.666667
```

(c) Below is the average treatment effect for the treated. This might differ from ATE because of a few reasons. First it could be random chance, as our selection here is quite small. Second it could be selection bias, or selection bias If the treated group is not the same as the untreated group, you would expect different.

```
ATT <- mean(table$Unit_treatment[table$D == 1])
ATT
```

```
[1] 2.333333
```

(d) Below is the average effect for the untreated.

```
ATU <- mean(table$Unit_treatment[table$D == 0])
ATU
```

```
[1] 1
```

(e) The naive average treatment effect is the following

$$NATE = ATE + E[Y_0|D = 1] - E[Y_0|D = 0] + (1 - P(D = 1))(ATT - ATU)$$

Breaking this down into its component parts, lets start with selection bias ($E[Y_0|D = 1] - E[Y_0|D = 0]$).

```
Selection_Bias = mean(table$Y0[table$D == 1]) - mean(table$Y0[table$D == 0])
Selection_Bias
```

```
[1] -0.6666667
```

Next, we calculate the Heterogeneous Treatment Effect Bias. This is 1 minus the probability of D = 1 (binary variable, can be found by taking average of column) multiplied by the difference between ATT and ATU:

```
Heterogeneous_TE_Bias = (1 - (mean(table$D) * (ATT - ATU)))
Heterogeneous_TE_Bias
```

```
[1] 0.3333333
```

Finally we put it all together, to calculate $NATE$

```
NATE = ATE + Selection_Bias + Heterogeneous_TE_Bias
NATE
```

```
[1] 1.333333
```

The difference between the $NATE$ and ATE:

```
Overall_bias = NATE - ATE
Overall_bias
```

```
[1] -0.3333333
```

(g) $NATE = ATE$ in the absence of selection bias or Heterogeneous Treatment Effect Bias. While individual outcomes of alternate realities are impossible to know, if the two populations are on average the same (ie, no bias in selection nor Heterogeneous TE bias), then $NATE = ATE$

For the NATE ($NATE = E[Y_1|D = 1] - E[Y_0|D = 0]$) to equal ATE:

$$(Y_1, Y_0) \perp\!\!\!\perp D$$

This means that selection bias will have to equal zero:

$$E[Y_0|D = 1] - E[Y_0|D = 0] = 0$$

And the Heterogeneous Treatment Effect Bias must also be equal to zero, or:

$$ATT - ATU = 0$$

So if treatment is independent of potential outcomes, then:

$$\frac{1}{N_T}\sum_{i=1}^{n}(y_i|d_i = 1) - \frac{1}{N_C}\sum_{i=1}^{n}(y_i|d_i = 0) = E[Y_1] - E[Y_0]$$
$$NATE = ATE$$