Problem Set 6: Prediction Modeling

Statistics 139 Teaching Staff

Due: November 03, 2023

This assignment is **due Friday, November 3 at 11:59pm**, handed in on Gradescope (remember, there are two separate submissions, one for your pdf, and another for you rmd file). Show your work and provide clear, explanations when asked. **Incorporate the relevant R output in this R markdown file**. Only the key output should be displayed for each problem and the relevant parts should be **highlighted** in some way. Make sure that you write-up any interpretation of R-code in your own words (don't just provide the output).

Collaboration policy (for this and all future problem sets): You are encouraged to discuss the problems with other students, but you must write up your solutions yourself and in your own words. Copying someone else's solution, or just making trivial changes is not acceptable.

Problem 1.

X_1, X_2, and X_3 are three explanatory variables in a multiple regression with $n = 28$ cases. The following table shows the residual sum of squares and degrees of freedom for all models (note: this table is in the file `ABC.csv` to facilitate using R to do the calculations):

Model Variables	Residual sum of squares	Degrees of freedom
None	8,100	27
X_1	6,240	26
X_2	5,980	26
X_3	6,760	26
X_1, X_2	5,500	25
X_1, X_3	5,250	25
X_2, X_3	5,750	25
X_1, X_2, X_3	5,160	24

(a) Calculate 3 statistics for each model: the estimate of σ^2, AIC, and BIC.

```
# load data
abc <- read.csv("data/abc.csv")
colnames(abc) <- c("var", "RSS", "df")
n <- 28

# calculate
abc$sigma.sq = abc$RSS/n
abc$AIC = n*log(abc$RSS/n) + 2*(n-abc$df-1)
abc$BIC = n*log(abc$RSS/n) + (n-abc$df-1)*log(n)

abc
```

```
##      var  RSS df sigma.sq      AIC      BIC
```

```
## 1  None  8100 27 289.2857 158.6876 158.6876
## 2     x1 6240 26 222.8571 153.3829 154.7151
## 3     x2 5980 26 213.5714 152.1912 153.5234
## 4     x3 6760 26 241.4286 155.6241 156.9563
## 5   x1x2 5500 25 196.4286 151.8484 154.5128
## 6   x1x3 5250 25 187.5000 150.5458 153.2102
## 7   x2x3 5750 25 205.3571 153.0930 155.7574
## 8 x1x2x3 5160 24 184.2857 152.0616 156.0583
```

(b) Summarize which model(s) is/are ranked best for each of the 3 statistics from part (a).

Model that includes X_1, X_2, X_3 has the lowest σ^2. Model that includes X_1 and X_3 has the lowest AIC (and is thus ranked best by this metric). Model that includes X_1 and X_3 has the lowest BIC (and is thus ranked best by this metric).

(c) Using the residual sum of squares, find the model indicated by forward selection. Start with the model 'None', and identify the single-variable model that has the smallest residual sum of squares, then perform an extra-sum-of-squares F-test to determine if that variable is significant. If it is, continue with the 2 predictor model. Continue until no more significant predictors can be added. Is this procedure guaranteed to find the "best" model (that is, where a determination of "best" is based on residual sum of squares)?

???

Problem 2.

What are risk factors for elevated blood pressure in the US (measured by systolic blood pressure, in mm Hg)? Several variables from the National Health and Nutrition Examination Survey (NHANES) are stored in `nhanes.csv`.

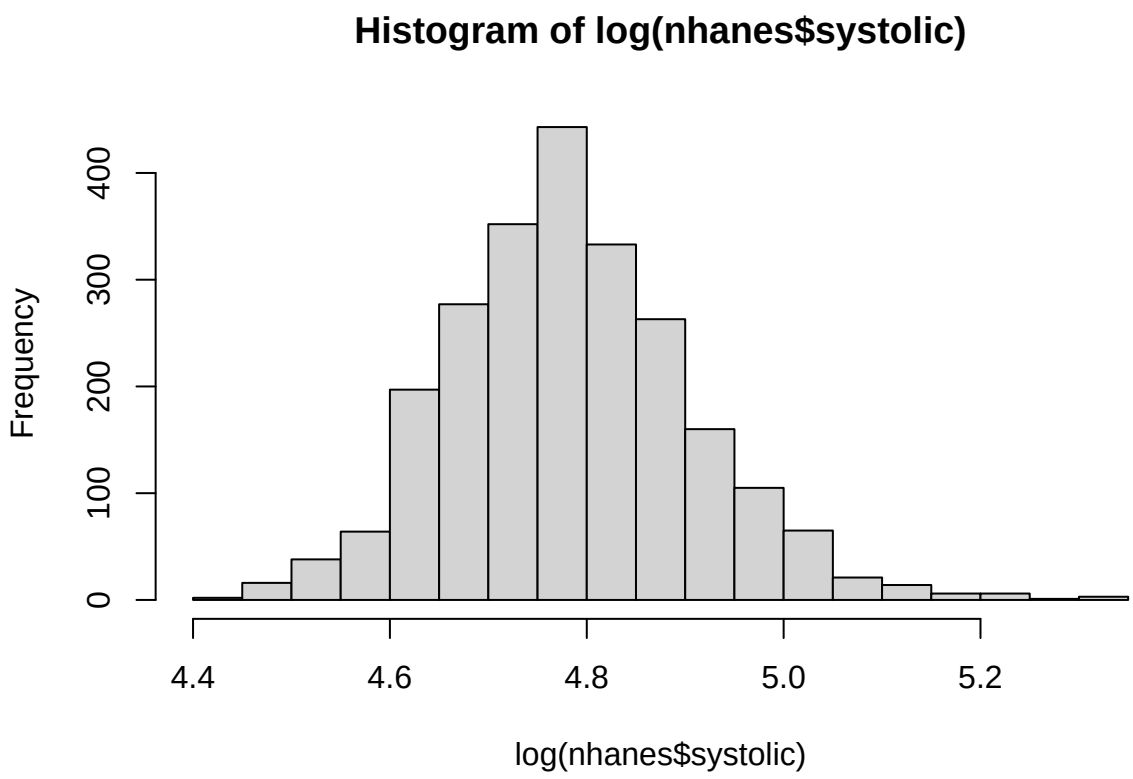
Descriptions of the variables are included below.

- `systolic`: systolic blood pressure, measured in mm Hg.
- `workhours`: self-reported number of hours in a typical work week.
- `jobtype`: description of job/work situation. The codes `1` through `5` correspond to an employee of a private company/individual for wages or salary, a federal government employee, a state government employee, a local government employee, or self-employed.
- `smoke`: coded `1` if the participant smokes regularly, `0` otherwise.
- `sleep`: self-reported number of hours study participant usually gets on weeknights or workdays; reported for participants aged 16 years or older.
- `active`: coded `1` if participant does moderate or vigorous intensity sports, fitness, or recreational activities; reported for participants 12 years or older.
- `diabetes`: coded `1` if the participant was told by a health professional that they have diabetes, `0` otherwise.
- `alcohol`: coded `1` if the participant has consumed at least 12 drinks of any type of alcoholic beverage in any one yer; reported for participants aged 18 years or older
- `female`: coded `1` if the participant is female, `0` otherwise.
- `age`: age in years at screening. Subjects 80 years or older were recorded as 80 years of age.
- `poverty`: a ratio of family income to poverty guidelines. Smaller numbers indicate more poverty; i.e., a number below 1 indicates income below the poverty level.
- `married`: marital status of study participant; reported for participants aged 20 or older. The codes `1` through `6` correspond to married, widowed, divorced, separated, never married, or living with partner.
- `education`: highest educational level of study participant, reported for participants aged 20 years or older. The codes `1` through `5` correspond to 8th grade, 9 to 11th grade, high school, some college, or college graduate.
- `race`: reported race of study participant: 1 = Mexican, 2 = Hispanic, 3 = White, 4 = Black, 6 = Asian, or 7 = Other.

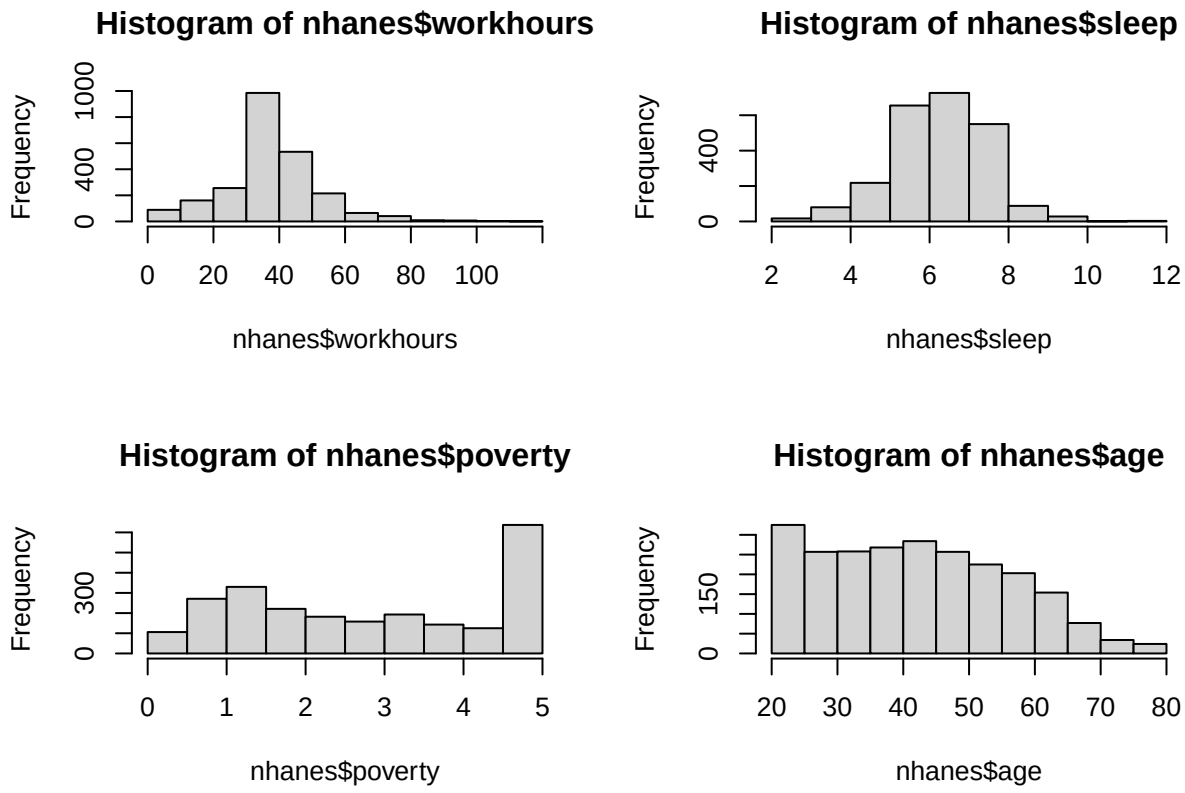
- `foreignborn`: coded `1` if participant was not born in the US, `0` otherwise
- `heartrate`: 60 second pulse rate
- `height`: standing height, measured in centimeters.
- `weight`: weight, measured in kilograms.
- `waist`: waist circumference, measured in centimeters.
- `bmi`: body mass index

(a) Explore the data graphically and decide whether the outcome variable (`systolic`) or any predictor variable(s) need to be transformed. Make sure you define any categorical variables as factors in R. If you decide to transform the response, use this transformed version as the response/outcome variable for all future models.

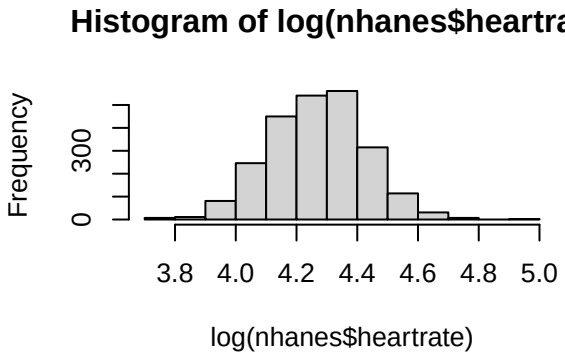
```
nhanes <- read.csv("data/nhanes.csv")

cols <- c("jobtype", "smoke", "active", "diabetes", "alcohol", "female", "married", "educ", "race", "for

for(i in 1:length(cols)){
  nhanes[[cols[i]]] <- as.factor(nhanes[[cols[i]]])
}

# we should log transform the outcome var
hist(log(nhanes$systolic), breaks=20)
```

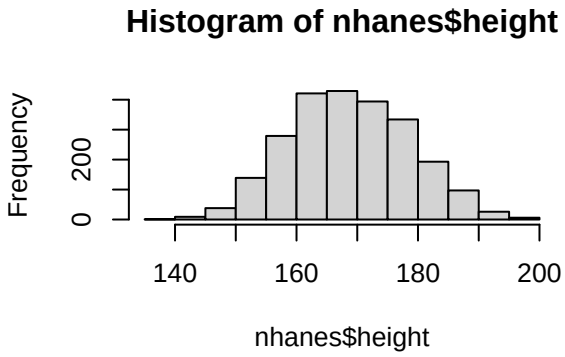


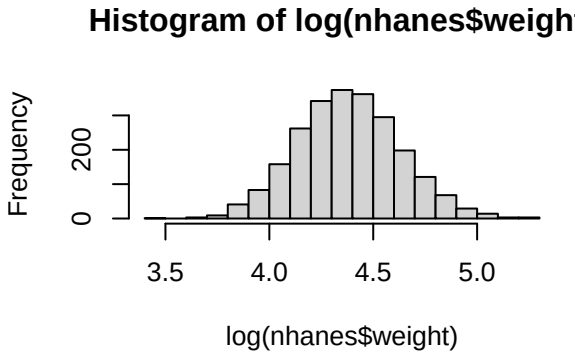
```r
nhanes$t.systolic <- log(nhanes$systolic)

# peek
par(mfrow=c(2,2))
hist(nhanes$workhours)
hist(nhanes$sleep)
hist(nhanes$poverty)
hist(nhanes$age)
```

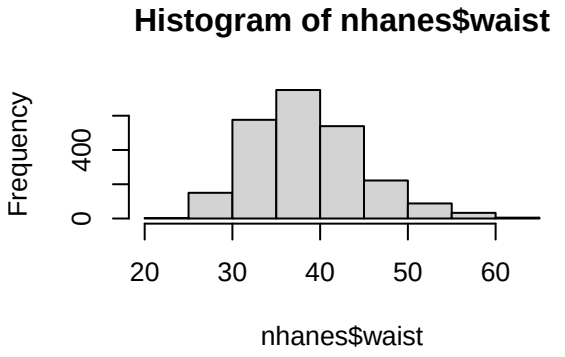



```r
hist(log(nhanes$heartrate))
hist(nhanes$height)
hist(log(nhanes$weight))
hist(nhanes$waist)
```

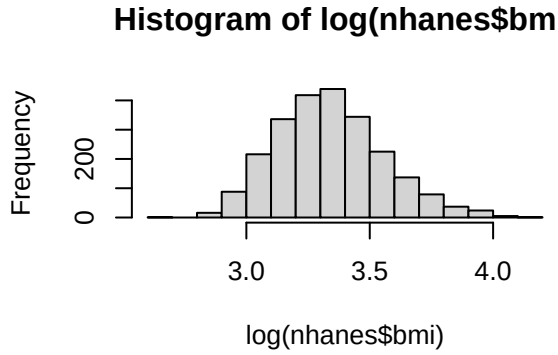









```r
hist(log(nhanes$bmi))

# transform
nhanes$t.heartrate <- log(nhanes$heartrate)
nhanes$t.weight    <- log(nhanes$weight)
nhanes$t.bmi       <- log(nhanes$bmi)

df.nhanes <- subset(nhanes, select=c(cols, "workhours", "sleep", "poverty", "age", "height", "waist", "t
```



(b) Fit a model with 'main effects' of all available predictors in their transformed states (call this **model1**). Your model should have 2332 degrees of freedom associated with the residuals (unless you used exotic transformations). Identify significant predictors (ignoring multiple comparisons).

```
model1 <- lm(t.systolic ~ ., df.nhanes)
summary(model1)
```

```
##
## Call:
## lm(formula = t.systolic ~ ., data = df.nhanes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41177 -0.07051 -0.00525  0.06618  0.52085
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.5621726  0.5314494   8.584  < 2e-16 ***
## jobtype2      0.0109843  0.0168777   0.651  0.51523
## jobtype3      0.0166668  0.0122034   1.366  0.17215
## jobtype4     -0.0052757  0.0095620  -0.552  0.58118
## jobtype5     -0.0041100  0.0098573  -0.417  0.67675
## smoke1        0.0022722  0.0051019   0.445  0.65609
## active1       0.0028312  0.0046991   0.603  0.54689
## diabetes1     0.0083365  0.0081261   1.026  0.30505
```

```
## alcohol1      0.0016551  0.0060714   0.273  0.78518
## female1      -0.0550569  0.0068308  -8.060 1.20e-15 ***
## married2      0.0258124  0.0159686   1.616  0.10613
## married3      0.0061575  0.0078334   0.786  0.43191
## married4      0.0180320  0.0144768   1.246  0.21304
## married5      0.0167417  0.0066338   2.524  0.01168 *
## married6      0.0035959  0.0089094   0.404  0.68654
## educ2        -0.0055868  0.0132498  -0.422  0.67332
## educ3        -0.0075946  0.0124284  -0.611  0.54122
## educ4        -0.0191229  0.0124328  -1.538  0.12416
## educ5        -0.0298945  0.0131402  -2.275  0.02299 *
## race2         0.0054238  0.0100559   0.539  0.58969
## race3         0.0066311  0.0085362   0.777  0.43734
## race4         0.0390674  0.0093094   4.197 2.81e-05 ***
## race6         0.0172664  0.0102662   1.682  0.09273 .
## race7         0.0256161  0.0139993   1.830  0.06741 .
## foreignborn1 -0.0021757  0.0072599  -0.300  0.76444
## workhours    -0.0001576  0.0001654  -0.952  0.34098
## sleep         0.0031459  0.0018571   1.694  0.09041 .
## poverty      -0.0016817  0.0017151  -0.980  0.32695
## age           0.0031678  0.0002232  14.195  < 2e-16 ***
## height       -0.0031594  0.0063790  -0.495  0.62045
## waist         0.0022172  0.0011983   1.850  0.06439 .
## t.heartrate   0.0484944  0.0154139   3.146  0.00168 **
## t.weight      0.2051975  0.5394497   0.380  0.70370
## t.bmi        -0.1692220  0.5408463  -0.313  0.75440
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1104 on 2332 degrees of freedom
## Multiple R-squared:  0.2335, Adjusted R-squared:  0.2226
## F-statistic: 21.52 on 33 and 2332 DF,  p-value: < 2.2e-16
```

According to the model output summary above, the significant predictors include `female1` (being female), `married5` (never married), `educ5` (college graduate), `race4` (Black), `age`, and `t.heartrate`.

(c) Use the **backward** variable selection procedure based on AIC to build a prediction model for `systolic` (transformed appropriately), starting from a model with all main effects. You may find the function `step()` helpful. There is no need to report the intermediate output of the `step()` function in your write-up, just report this model's coefficient estimates (not the full `summary` output), R^2, and AIC. Call the resulting model **model2**.

The function `lm()` has some useful shortcuts for specifying formulas. For example, to include the main effects of all variables in the dataset `MyData`:

```
# to include the main effects of all variables in the dataset \texttt{MyData}
# lm(y ~ ., data = MyData)

# To include all variables and their pairwise interactions:
# lm(y ~ .^2, data = MyData)

# build model
model2 <- step(model1, direction="backward", trace=0)
```

```r
# pull outputs why is AIC negative????
model2$coefficients
```

```
##   (Intercept)       female1      married2      married3      married4
##  4.4711867531 -0.0554488223  0.0285840583  0.0078838671  0.0201425488
##      married5      married6         educ2         educ3         educ4
##  0.0178531502  0.0054414503 -0.0042329010 -0.0068378966 -0.0186509180
##         educ5         race2         race3         race4         race6
## -0.0312301010  0.0050677480  0.0071656179  0.0417077806  0.0151664816
##         race7         sleep           age        height         waist
##  0.0264392480  0.0031786190  0.0031184212 -0.0009351264  0.0034715176
##   t.heartrate
##  0.0473192100
```

```r
summary(model2)$r.squared
```

```
## [1] 0.2308766
```

```r
AIC(model2)
```

```
## [1] -3694.823
```

(d) Next, run a **forward** variable selection procedure starting with **model2**, with the upper scope for the final model set to include all the two-way interaction terms for the variables in **model2**. Report this model's coefficient estimates, R^2, and AIC. Call this **model3**.

Note: The predictors from **model2** can be printed in a list in R via the command `model2$terms[[3]]`. This forward variable selection can be performed using the `step()` function as follows, where `interactionModel` is the `lm` fit with all variables from **model2** and their interactions:

```r
# model2$terms[[3]]

interactionModel <- lm(t.systolic ~ (female + married + educ + race + sleep + age + height + waist + t.h

model3 <- step(model2,
     scope = list(upper = formula(interactionModel)),
     direction = "forward",
     trace=0)

model3$coefficients
```

```
##       (Intercept)           female1          married2          married3
##      3.226305e+00     -3.686678e-02      3.948803e-03      7.735710e-02
##          married4          married5          married6             educ2
##     -1.441456e-01      5.592995e-02      3.293121e-02      4.207470e-01
##             educ3             educ4             educ5             race2
##      8.388744e-01      1.034288e+00      7.011704e-01      2.729233e-01
##             race3             race4             race6             race7
##     -3.132011e-01     -2.234106e-01      1.933576e-01     -2.354427e-01
##             sleep               age            height             waist
##      5.758888e-03      2.952217e-02      2.979170e-03     -1.095434e-02
```

```
##       t.heartrate        age:height      married2:age      married3:age
##      1.787852e-01     -9.148123e-05      2.911541e-05     -1.533756e-03
##      married4:age      married5:age      married6:age educ2:t.heartrate
##      3.458650e-03     -1.166763e-03     -7.084108e-04     -9.897499e-02
## educ3:t.heartrate educ4:t.heartrate educ5:t.heartrate race2:t.heartrate
##     -1.980288e-01     -2.462945e-01     -1.709339e-01     -6.279415e-02
## race3:t.heartrate race4:t.heartrate race6:t.heartrate race7:t.heartrate
##      7.451033e-02      6.225179e-02     -4.192058e-02      6.063955e-02
##   age:t.heartrate     female1:sleep waist:t.heartrate       female1:age
##     -2.572699e-03     -6.371740e-03      3.399936e-03      6.389468e-04
```

```
summary(model3)$r.squared
```

```
## [1] 0.2620449
```

```
AIC(model3)
```

```
## [1] -3754.7
```

(e) Finally, use a combined **stepwise** procedure to perform model selection. Start with a model with all main effects and specify the intercept-only model (**model0**) as a lower limit model and a full model including all two-way interactions of *all* possible predictor variables as the upper-limit as shown below (call this the **fullInteractionModel**). Report this model's coefficient estimates, R^2, and AIC. Call this **model4**. Note, this may take a minute or two...the use of the code chunk option `cache=TRUE` can be helpful so you do not need to wait for this to run every time you want to knit the file into a pdf.

```
model0 <- lm(t.systolic ~ 1, df.nhanes)
fullInteractionModel <- lm(t.systolic ~ .^2, df.nhanes)

model4 <- step(model1, scope = list(lower = formula(model0),
                                    upper = formula(fullInteractionModel)),
               direction = "both", trace=0)

model4$coefficients
```

```
##             (Intercept)                jobtype2                jobtype3
##            4.322807e+00            5.055602e-01            1.260996e+00
##                jobtype4                jobtype5                 active1
##            4.221436e-01            1.106765e-01            3.668908e-01
##               diabetes1                 alcohol1                  female1
##           -5.101115e-01           -1.860721e-01           -7.315132e-02
##                married2                married3                married4
##            2.967359e-01            1.948376e-01            4.710222e-01
##                married5                married6                   educ2
##            7.304179e-02            2.955002e-01            6.455630e-01
##                   educ3                   educ4                   educ5
##            1.072277e+00            1.209040e+00            7.693912e-01
##                   race2                   race3                   race4
##           -3.604843e-01           -1.726475e-01            5.535068e-03
##                   race6                   race7              foreignborn1
##           -6.633670e-01           -2.095269e-01            3.761416e-01
##               workhours                   sleep                 poverty
```

```
##           -6.203996e-03            8.310345e-03           -5.489448e-03
##                     age                  height                   waist
##            4.005924e-02           -6.141529e-03           -3.026134e-03
##             t.heartrate                t.weight                   t.bmi
##            4.541954e-01            3.763447e-01           -1.092271e+00
##              age:height       foreignborn1:age        alcohol1:female1
##           -8.643137e-05            1.000659e-03           -2.807591e-02
##    jobtype2:t.heartrate    jobtype3:t.heartrate    jobtype4:t.heartrate
##           -1.113470e-01           -2.699611e-01           -1.080569e-01
##    jobtype5:t.heartrate       jobtype2:alcohol1       jobtype3:alcohol1
##            3.148815e-03            6.354019e-02           -9.952754e-02
##       jobtype4:alcohol1       jobtype5:alcohol1            married2:age
##            9.327051e-03            3.406134e-02           -2.433751e-03
##            married3:age            married4:age            married5:age
##           -1.083823e-03            2.132101e-03           -1.191852e-03
##            married6:age        married2:poverty        married3:poverty
##           -2.199179e-04            2.675380e-02           -2.268124e-04
##        married4:poverty        married5:poverty        married6:poverty
##           -3.819403e-04            1.186435e-02            5.289409e-03
##             race2:t.bmi             race3:t.bmi             race4:t.bmi
##            1.088697e-01            5.251677e-02            1.066734e-02
##             race6:t.bmi             race7:t.bmi          t.weight:t.bmi
##            2.081982e-01            6.861593e-02            1.058057e-01
##                age:t.bmi       educ2:t.heartrate       educ3:t.heartrate
##           -5.198434e-03           -1.521636e-01           -2.525460e-01
##       educ4:t.heartrate       educ5:t.heartrate foreignborn1:t.heartrate
##           -2.869053e-01           -1.848729e-01           -9.904708e-02
##     active1:t.heartrate             female1:age       active1:workhours
##           -7.342078e-02            9.728475e-04            8.344682e-04
##           jobtype2:sleep          jobtype3:sleep          jobtype4:sleep
##           -1.201854e-02           -1.899943e-03            4.822415e-03
##           jobtype5:sleep      diabetes1:married2      diabetes1:married3
##           -2.348516e-02            1.861601e-02           -1.779070e-02
##      diabetes1:married4      diabetes1:married5      diabetes1:married6
##            1.052397e-01            5.524370e-02            3.029342e-02
##             active1:age        diabetes1:waist            female1:sleep
##           -7.538797e-04           -3.235557e-03           -8.348996e-03
##       diabetes1:poverty        diabetes1:height         age:t.heartrate
##           -1.132272e-02            1.675140e-03           -2.618718e-03
##               age:waist       alcohol1:t.weight   diabetes1:t.heartrate
##            1.409738e-04            5.343349e-02            9.020855e-02
##           active1:waist       active1:diabetes1            alcohol1:age
##           -1.475727e-03            2.590578e-02           -5.624087e-04
##           female1:waist         workhours:t.bmi      married2:workhours
##            1.640995e-03            1.349158e-03           -3.299160e-03
##      married3:workhours      married4:workhours      married5:workhours
##            2.166050e-04            2.841323e-04            7.720345e-04
##      married6:workhours           workhours:age          married2:t.bmi
##           -3.553377e-04            2.681361e-05           -3.239355e-02
##          married3:t.bmi          married4:t.bmi          married5:t.bmi
##           -4.493873e-02           -1.768546e-01           -2.709331e-02
##          married6:t.bmi
##           -8.588998e-02
```

```r
summary(model4)$r.squared
```

```
## [1] 0.3205184
```

```r
AIC(model4)
```

```
## [1] -3824.02
```

(f) Select a best final model among 5 models based on their AICs: **model1** through **model4** and the **fullInteractionModel**. Perform a brief model check of assumptions on your selected model.

```r
AIC(model1)
```

```
## [1] -3676.832
```

```r
AIC(model2)
```

```
## [1] -3694.823
```

```r
AIC(model3)
```

```
## [1] -3754.7
```

```r
AIC(model4)
```

```
## [1] -3824.02
```

```r
AIC(fullInteractionModel)
```

```
## [1] -3456.804
```

(g) Use the model chosen in part (f) to interpret the association of systolic blood pressure with the variable `female`. If `female` is not in your chosen model in any fashion, interpret what that means.

Model 4 performs the best because it has the lowest AIC. ???

Problem 3.

Use cross-validation to compare three different models from the previous problem (as defined below) to predict your (transformed) systolic response variable.

1. **interactionModel** from 2(d)
2. **model3**
3. **model4**

For each of the three models, do the following:

- For 100 iterations, randomly select 2,000 observations on which to train each model, and keep the remaining observations as the validation set in each iteration.
- Predict both systolic blood pressure and log-transformed systolic blood pressure for all observations in the validation sets, and save the residual sums of squares of the validation set in each iteration (for the 2 versions of the response: log and untransformed).

In your solution, summarize your results as to which of the three models performs best in predicting systolic blood pressure and which performs best in predicting log-transformed systolic blood pressure. Be sure to address whether the choice of best model agree with your selection from the problem 2.

Note: if you were to actually then use the best model for future observations, you should always refit the chosen model on all the data sampled, not just one training set (here, of $n = 2000$).

```
nsims <- 100
rss <- data.frame(interactionModel = as.numeric(),
          m3 = as.numeric(),
          m4 = as.numeric(),
          best.model = as.numeric())
t.rss <- data.frame(interactionModel = as.numeric(),
          m3 = as.numeric(),
          m4 = as.numeric(),
          best.model = as.numeric())

for (i in 1:nsims){

  train.ids = sample(1:nrow(df.nhanes), 2000)
  train = df.nhanes[train.ids,]
  test = df.nhanes[-train.ids,]

  fit1 = lm(formula(interactionModel), data=train)
  fit2 = lm(formula(model3),           data=train)
  fit3 = lm(formula(model4),           data=train)

  # transformed rss
  t.rss[i,1] <- sum(test$t.systolic - predict(fit1, new=test))^2
  t.rss[i,2] <- sum(test$t.systolic - predict(fit2, new=test))^2
  t.rss[i,3] <- sum(test$t.systolic - predict(fit3, new=test))^2
  t.rss[i,4] <- which.min(t.rss[i, ])

  # untransformed rss
  rss[i,1] <- sum(exp(test$t.systolic) - exp(predict(fit1, new=test)))^2
  rss[i,2] <- sum(exp(test$t.systolic) - exp(predict(fit2, new=test)))^2
```

```
  rss[i,3] <- sum(exp(test$t.systolic) - exp(predict(fit3, new=test)))^2
  rss[i,4] <- which.min(rss[i, ])
}

table(t.rss[,"best.model"])
```

```
##
##  1  2  3
## 36 32 32
```

```
table(rss[,"best.model"])
```

```
##
##  1  2  3
## 37 26 37
```

Problem 4.

This problem is intended to investigate the optimal choice of train-validation splitting ratios in a model comparison problem. We will be comparing k-fold cross-validation for $k \in \{2, 10, 25\}$. For $n = 50$ (and eventually also for $n = 500$), create data based on the following data-generating process:

i) Sample $Z_1, Z_2, Z_3, X_1, ..., X_{10}$ all independently from the standard normal distribution.

ii) Sample $(Y|\vec{Z}, \vec{X}) \sim N(1 \cdot Z_1 + 3 \cdot Z_2 + 9 \cdot Z_3, 5^2)$ independently from each other.

*Note, it is more efficient to sample all Z_1, Z_2, Z_3 and $\vec{X}$ for each iteration from one `rnorm` function call, and then reorganize them into the separate variables for data creation and model fitting.

For each of `nsims=200` iterations, perform k-fold cross-validation (for each of the 3 choices of k mentioned above) to determine which of the 4 following models is the best out-of-sample prediction model separately for each choice of k:

1) `lm(Y ~ Z3)`
2) `lm(Y ~ Z2 + Z3)`
3) `lm(Y ~ Z1 + Z2 + Z3)`
4) `lm(Y ~ Z1 + Z2 + Z3 + X1 + X2 + ... + X10)`

In the end for each choice of k, you should determine which of the 4 models above is *best* for 200 iterations.

```
# defining your own simulation function could be useful but not required.
# but much sure your approach is working outside of the function before defining one
# my.sim = function(n = 50, nsims = 200, k = c(2,5,25), seed = NA)

# you simulation code here

data <- rnorm(13, 0, 1)
y <- rnorm(1, 1*data[1] + 3*data[2] + 9*data[3], 5)
```

(a) How often is each model selected as best (as measured by average validation set squared error)? Provide a 4 x 3 table that displays the number of times each model is chosen where the columns represent the choice of k and the rows represent each model mentioned above. Note: the columns should sum up to `nsims=200`.

(b) Interpret the results: is there a clear winner for the choice of k here? How often is each cross-validation choosing the *correct* model? Do they tend to favor the underfit or overfit models? Explain.

(c) Rerun the simulation above for $n = 500$. Provide the analogous 4 x 3 table to part (a) above.

(d) Compare the results for the $n = 50$ and $n = 500$ cases. Is there a clear winner for the choice of k when $n = 500$? How does sample size affect the choice of the *underfit*, *correct*, and *overfit* models (in this context)? Explain.

(e) What is missing/lacking in this small simulation study? How would you modify or extend this simulation to better *investigate the optimal choice of train-validation splitting ratios in a model comparison problem*? Explain your choice(s) in up to 8 sentences (do not implement these changes!!!).