Problem Set 6: Prediction Modeling

Linh Vu (Collab: Brice Laurent)

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.

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

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

abc
```

```
##       var  RSS df sigma.sq      AIC      BIC
## 1    None 8100 27      300 160.6876 162.0198
## 2      x1 6240 26      240 155.3829 158.0473
## 3      x2 5980 26      230 154.1912 156.8556
## 4      x3 6760 26      260 157.6241 160.2885
## 5    x1x2 5500 25      220 153.8484 157.8450
## 6    x1x3 5250 25      210 152.5458 156.5424
## 7    x2x3 5750 25      230 155.0930 159.0896
## 8  x1x2x3 5160 24      215 154.0616 159.3905
```

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

Model that includes X_1 and X_3 has the lowest σ^2, lowest AIC, and lowest BIC and is thus ranked best by all three metrics.

(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)?

```r
# ESS F-test comparing `none` and `x2`
fstat.1 <- ((abc$RSS[1] - abc$RSS[3])/1)/(abc$RSS[3]/(abc$df[3]))
1 - pf(fstat.1, 1, abc$df[3])
```

```
## [1] 0.005391745
```

```r
# ESS F-test comparing `x2` and `x1x2`
fstat.2 <- ((abc$RSS[3] - abc$RSS[5])/1)/(abc$RSS[5]/(abc$df[5]))
1 - pf(fstat.2, 1, abc$df[5])
```

```
## [1] 0.1521369
```

Starting with the model `None`, we proceed with the model `x2` because it is the single-variable model with the smallest RSS. Since the F test statistic is 9.217 and the p-value is 0.005, we reject the null hypothesis and conclude that adding X_2 significantly improves the model.

Now we proceed with model `x2`. We compare it with model `x1x2` because it is the double-variable model with the smallest RSS. Since the F test statistic is 2.182 and the p-value is 0.152, we retain the null hypothesis and conclude that adding X_1 does not improve the model, and we stop with model `x2`.

In part (a), we found that the model with the lowest RSS is `x1x2x3`, so this method doesn't guarantee that we end up with the "best" model. This is because forward selection can get us stuck with a locally optimal model (and perhaps few predictors) without considering all possible models, and the lowest-RSS method generally gives us the model with the highest number of predictors.

Problem 2.

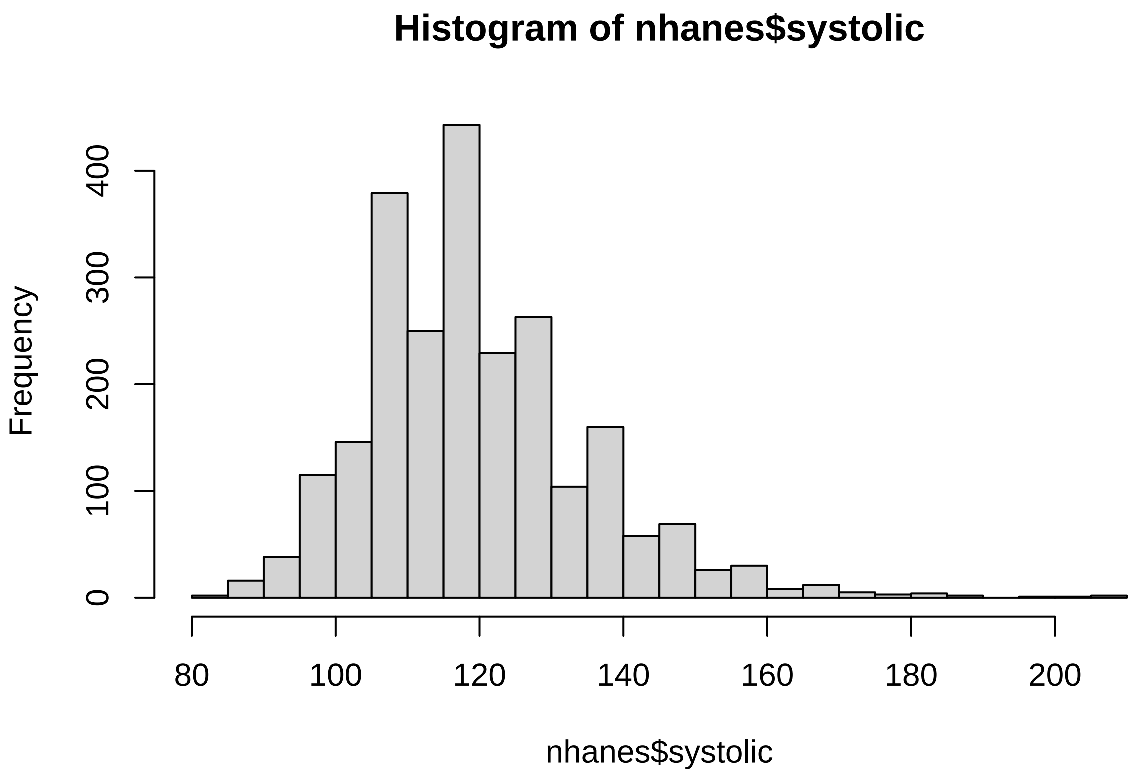
What are risk factors for elevated blood pressure in the US (measured by systolic blood pressure, in mm Hg)? Several variables from the National Health and Nutrition Examination Survey (NHANES) are stored in `nhanes.csv`.

Descriptions of the variables are included below.

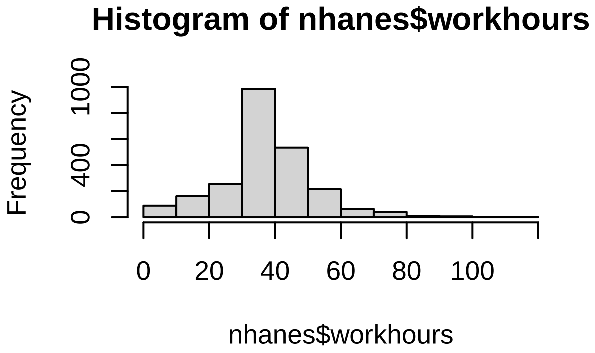
- `systolic`: systolic blood pressure, measured in mm Hg.
- `workhours`: self-reported number of hours in a typical work week.
- `jobtype`: description of job/work situation. The codes `1` through `5` correspond to an employee of a private company/individual for wages or salary, a federal government 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

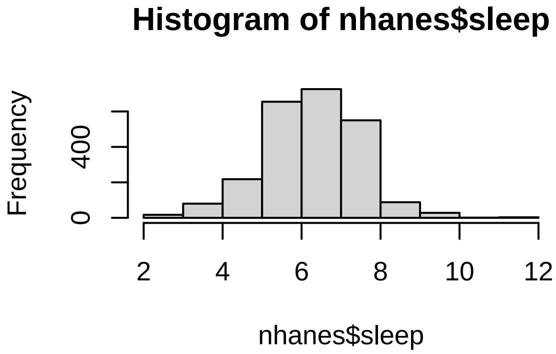
(a) Explore the data graphically and decide whether the outcome variable (`systolic`) or any predictor variable(s) need to be transformed. Make sure you define any categorical variables as factors in R. If you decide to transform the response, use this transformed version as the response/outcome variable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

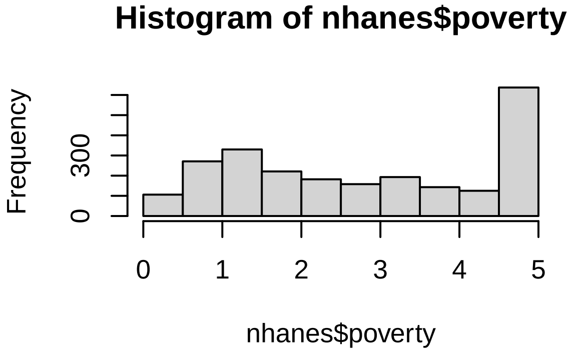
```r
nhanes <- read.csv("data/nhanes.csv")

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

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

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

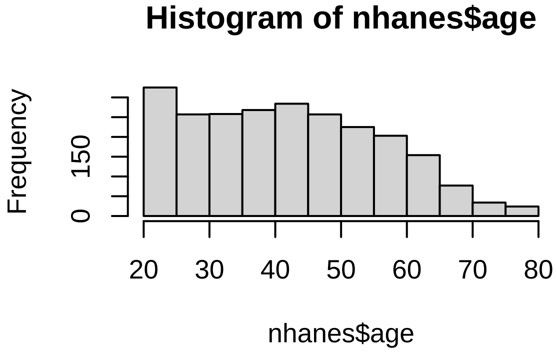


```
nhanes$t.systolic <- log(nhanes$systolic)

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

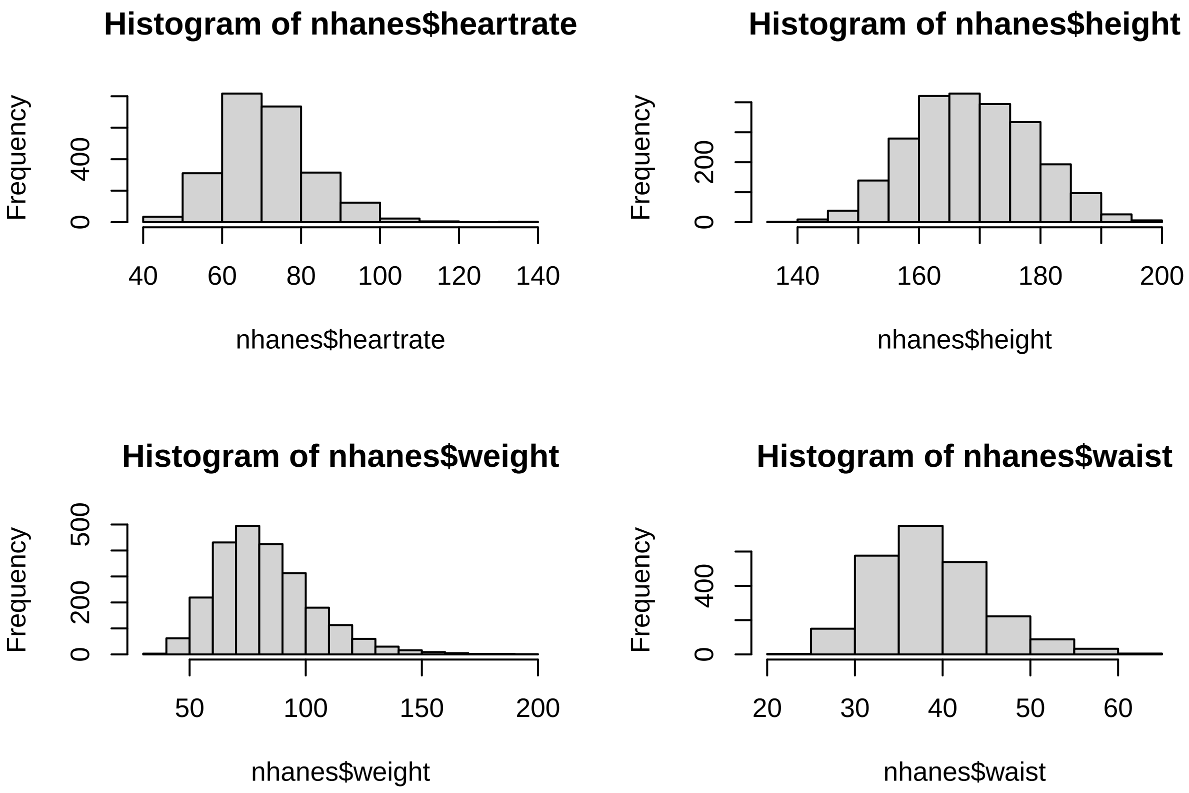




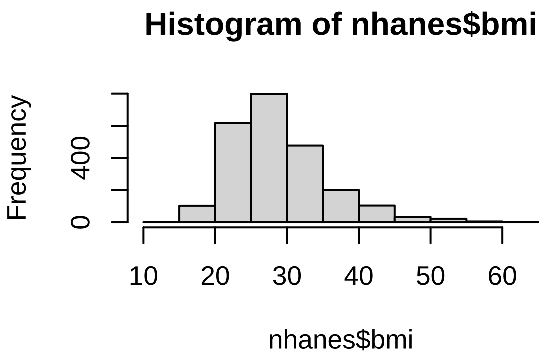




```r
hist(nhanes$heartrate)
hist(nhanes$height)
hist(nhanes$weight)
hist(nhanes$waist)
```



```
hist(nhanes$bmi)

df.nhanes <- subset(nhanes, select=-c(systolic, id))
```



I decided to log transform the response variable because the untransformed version is right-skewed. I also changed the categorical variables from numeric into factor variables.

(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
## -0.41291 -0.07090 -0.00554  0.06655  0.51900
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.6445232  0.1844429  25.181  < 2e-16 ***
## workhours     -0.0001533  0.0001654  -0.927  0.35414
## jobtype2       0.0107872  0.0168729   0.639  0.52268
## jobtype3       0.0164363  0.0122010   1.347  0.17807
## jobtype4      -0.0051276  0.0095551  -0.537  0.59157
## jobtype5      -0.0042585  0.0098563  -0.432  0.66574
## smoke1         0.0025012  0.0050975   0.491  0.62370
## sleep          0.0031363  0.0018563   1.689  0.09126 .
## active1        0.0029155  0.0046941   0.621  0.53459
## diabetes1      0.0082411  0.0081245   1.014  0.31052
## alcohol1       0.0019281  0.0060481   0.319  0.74991
## female1       -0.0556098  0.0067959  -8.183 4.51e-16 ***
## age            0.0032101  0.0002265  14.172  < 2e-16 ***
## poverty       -0.0016973  0.0017153  -0.989  0.32254
## married2       0.0259908  0.0159589   1.629  0.10353
## married3       0.0060270  0.0078299   0.770  0.44153
## married4       0.0178299  0.0144768   1.232  0.21821
## married5       0.0165496  0.0066227   2.499  0.01252 *
## married6       0.0034518  0.0089067   0.388  0.69839
## educ2         -0.0051196  0.0132293  -0.387  0.69880
## educ3         -0.0073253  0.0124101  -0.590  0.55507
## educ4         -0.0188042  0.0124177  -1.514  0.13008
## educ5         -0.0295581  0.0131136  -2.254  0.02429 *
## race2          0.0050937  0.0100647   0.506  0.61284
## race3          0.0064374  0.0085413   0.754  0.45112
## race4          0.0382169  0.0093343   4.094 4.38e-05 ***
## race6          0.0166080  0.0102583   1.619  0.10558
## race7          0.0248702  0.0140283   1.773  0.07638 .
## foreignborn1 -0.0019199  0.0072647  -0.264  0.79159
## heartrate      0.0006739  0.0002100   3.208  0.00135 **
## height        -0.0009058  0.0010769  -0.841  0.40035
## weight         0.0002492  0.0010493   0.238  0.81227
## waist          0.0016663  0.0012379   1.346  0.17841
## bmi            0.0009927  0.0029900   0.332  0.73991
## ---
## 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:  0.2338, Adjusted R-squared:  0.223
## F-statistic: 21.57 on 33 and 2332 DF,  p-value: < 2.2e-16
```

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

# pull outputs
model2$coefficients
```

```
##   (Intercept)         sleep       female1           age      married2
##  4.5948777686  0.0032180743 -0.0553241511  0.0032276839  0.0284499796
##      married3      married4      married5      married6         educ2
##  0.0073884206  0.0198817320  0.0180830913  0.0049960236 -0.0043056265
##         educ3         educ4         educ5         race2         race3
## -0.0071154139 -0.0189541900 -0.0318890313  0.0052693289  0.0070413071
##         race4         race6         race7     heartrate        height
##  0.0393127284  0.0159208292  0.0255537126  0.0006843401 -0.0006812026
##         waist           bmi
##  0.0017932858  0.0016168480
```

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

```
## [1] 0.2316294
```

```
AIC(model2)
```

```
## [1] -3695.14
```

The AIC of `model2` is -3695.1395902, and the R^2 is 0.232.

(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:

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

interactionModel <- lm(t.systolic ~ (sleep + female + age + married + educ + race
                                     + heartrate + height + waist + bmi)^2,
                       df.nhanes)

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

model3$coefficients
```

```
##     (Intercept)           sleep         female1             age        married2
##    3.728114e+00    5.851784e-03   -3.323331e-02    2.321945e-02    2.488245e-02
##        married3        married4        married5        married6           educ2
##    7.534694e-02   -1.520471e-01    4.571519e-02    2.003871e-02    7.491097e-02
##           educ3           educ4           educ5           race2           race3
##    1.689597e-01    2.056283e-01    1.206819e-01   -8.121572e-03   -1.037923e-01
##           race4           race6           race7       heartrate          height
##   -2.435375e-02   -1.203991e-01   -8.095773e-02    3.784887e-03    3.746063e-03
##           waist             bmi      age:height    age:married2    age:married3
##   -5.500668e-03    4.533427e-03   -1.080583e-04   -2.995994e-04   -1.474775e-03
##    age:married4    age:married5    age:married6         age:bmi       race2:bmi
##    3.585843e-03   -9.402614e-04   -4.210155e-04   -1.904319e-04    2.571799e-03
##       race3:bmi       race4:bmi       race6:bmi       race7:bmi educ2:heartrate
##    9.151300e-04   -5.790463e-05    6.708579e-03    5.259215e-04   -1.045644e-03
## educ3:heartrate educ4:heartrate educ5:heartrate race2:heartrate race3:heartrate
##   -2.399628e-03   -3.067742e-03   -2.038250e-03   -8.697317e-04    1.094807e-03
## race4:heartrate race6:heartrate race7:heartrate   sleep:female1   age:heartrate
##    9.244955e-04   -5.514211e-04    1.166450e-03   -7.131295e-03   -3.173380e-05
##       age:waist       waist:bmi     female1:age
##    1.488296e-04    7.345356e-05    7.053332e-04
```

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

```
## [1] 0.2694634
```

```r
AIC(model3)
```

```
## [1] -3762.606
```

The AIC of `model2` is -3762.605672, and the R^2 is 0.269.

(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.

```r
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
```

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

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

```
AIC(model4)
```

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

The AIC of `model4` is -3824.0201415, and the R^2 is 0.321.

(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.

```
# get AIC
AIC(model1)
```

```
## [1] -3677.935
```

```
AIC(model2)
```

```
## [1] -3695.14
```

```
AIC(model3)
```

```
## [1] -3762.606
```

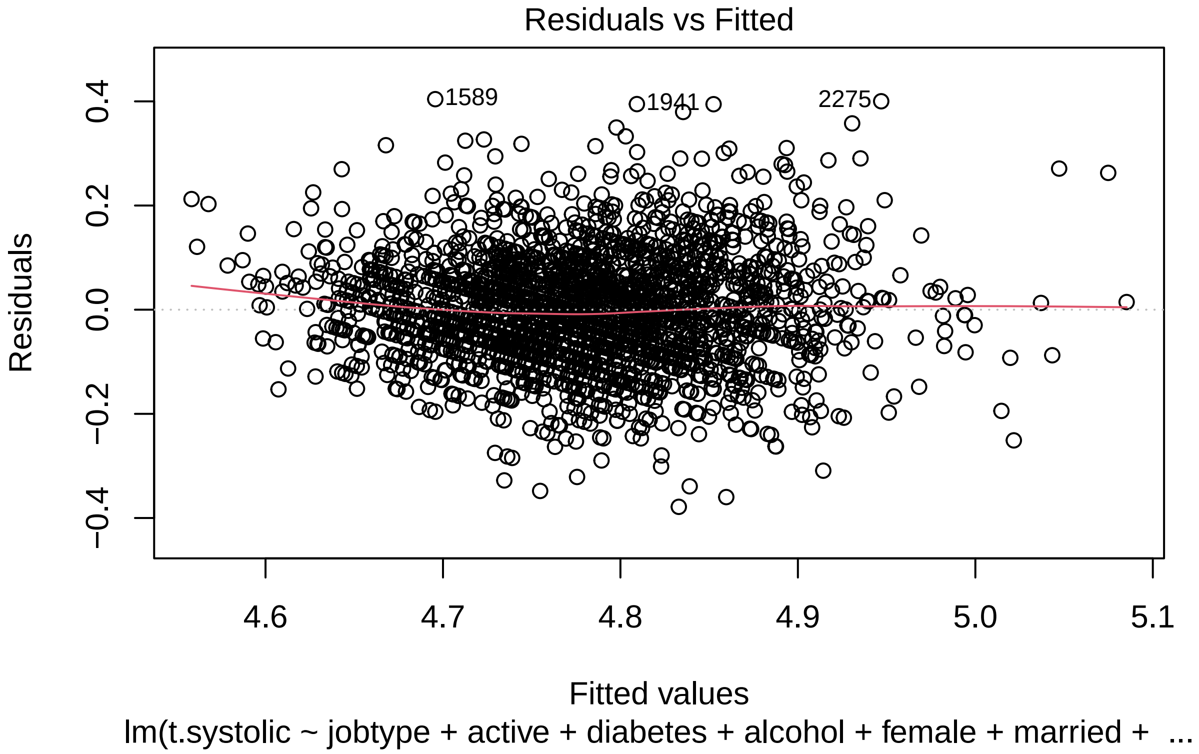
```
AIC(model4)
```

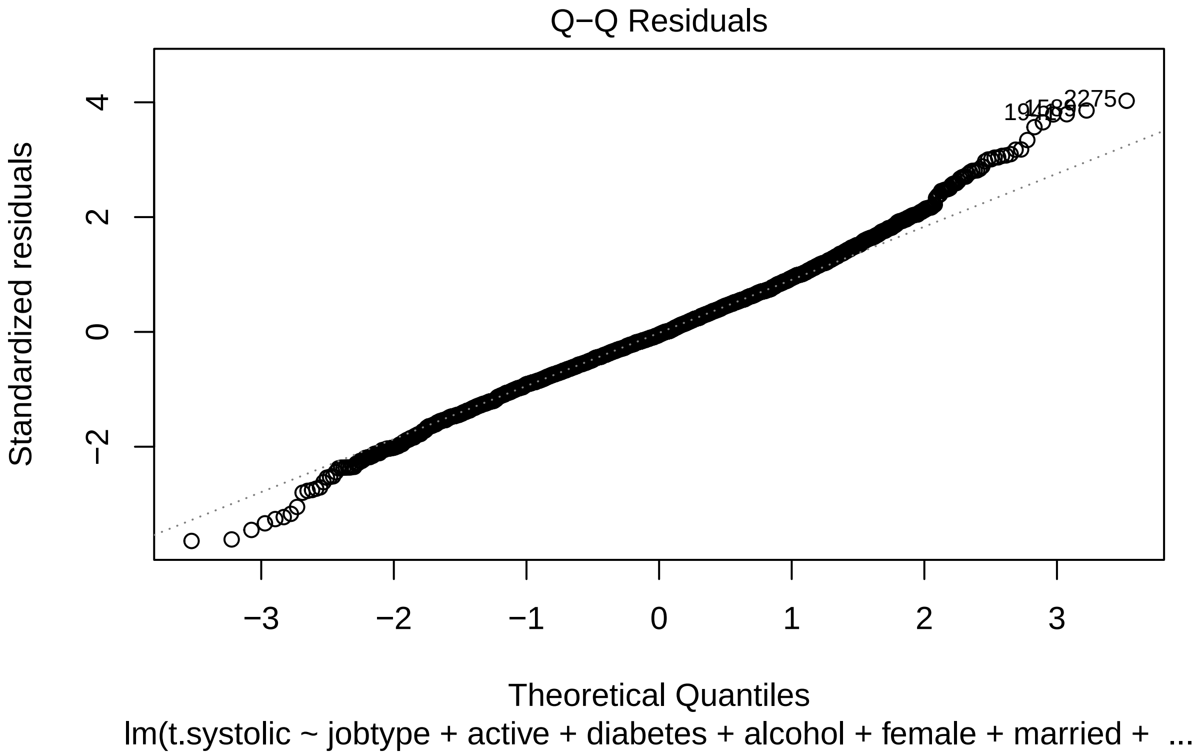
```
## [1] -3824.02
```

```
AIC(fullInteractionModel)
```

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

```
# check assumptions
plot(model4, which=c(1,2))
```


Residuals vs Fitted
Residuals
0.4
0.2
0.0
−0.2
−0.4
1589
1941
2275
4.6
4.7
4.8
4.9
5.0
5.1
Fitted values
lm(t.systolic ~ jobtype + active + diabetes + alcohol + female + married + ...


Q−Q Residuals
Standardized residuals
4
2
0
−2
1941
1589
2275
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(t.systolic ~ jobtype + active + diabetes + alcohol + female + married + ...

The best model, which has the lowest AIC, is `model4`. In terms of assumptions, the linearity assumption is satisfied because the points are equally distributed above and below the horizontal line (per plot 1). The homoskedasticity assumption is also met; there doesn't seem to be any fanning/funneling pattern (per plot 1). The normality assumption is also met because the standardized quantiles of the residuals closely resemble the theoretical quantiles in the QQ plot. The independence assumption is met depending on the design of the study (eg. if subjects are related and if the condition is genetic-based, then the independence assumption might not be met).

(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.

In `model4`, the coefficient estimate for `female1` is -0.108. This means that holding all other predictors at 0, being female decreases the systolic blood pressure by 0.108 mm Hg compared to being male.

The coefficient estimate for `alcohol1:female1` is -0.0264. This means that holding all other interaction predictors (except for alcohol) at 0, when a person drinks, the difference in systolic blood pressure of an average female individual vs an average male individual is $-0.108 - 0.0264 = -0.1344$ mm Hg.

The coefficient estimate for `female1:age` is 0.00114. This means that holding all other interaction predictors (except for age) at 0, when age increases by 1, the difference in systolic blood pressure of an average female individual vs an average male individual becomes $-0.108 + 0.00114 = -0.10686$ mm Hg.

The coefficient estimate for `sleep:female1` is -0.00735. This means that holding all other interaction predictors (except for sleep) at 0, when hours of sleep increase by 1, the difference in systolic blood pressure of an average female individual vs an average male individual becomes $-0.108 - 0.00735 = -0.11535$ mm Hg.

The coefficient estimate for `female1:poverty` is -0.00444. This means that holding all other interaction predictors (except for poverty) at 0, when the poverty index increases by 1 unit (person becomes more well off), the difference in systolic blood pressure of an average female individual vs an average male individual becomes $-0.108 - 0.00444 = -0.11244$ mm Hg.

The coefficient estimate for `female1:waist` is 0.00254. This means that holding all other interaction predictors (except for waist) at 0, when the waist circumference increases by 1 cm, the difference in systolic blood pressure of an average female individual vs an average male individual becomes $-0.108 + 0.00254 = -0.10546$ mm Hg.

The mean difference between systolic blood pressure between female and male individuals is $-0.108 - 0.0264a + 0.001140b - 0.00735c - 0.00444d + 0.00254e$ mmHg, where (a, b, c, d, e) are the interaction terms between female 1 and (alcohol1, age, sleep, poverty, waist), respectively, while holding all the other coefficient estimates and interaction estimates constant. If we consider two average individuals who, say, both drink (alcohol=1), are 42 years old, sleep 7 hours a night, have a poverty index of 2.8, and have a waist of 38cm, and other factors being equal, the difference in systolic blood pressure between the female and male individual is -0.05 mmHg, per the calculation below (i.e. the male individual has higher systolic bp on average).

```
-0.108 - 0.0264*(1) + 0.001140*mean(df.nhanes$age) - 0.00735*mean(df.nhanes$sleep) - 0.00444*mean(df.nha
```

```
## [1] -0.0500767
```

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.

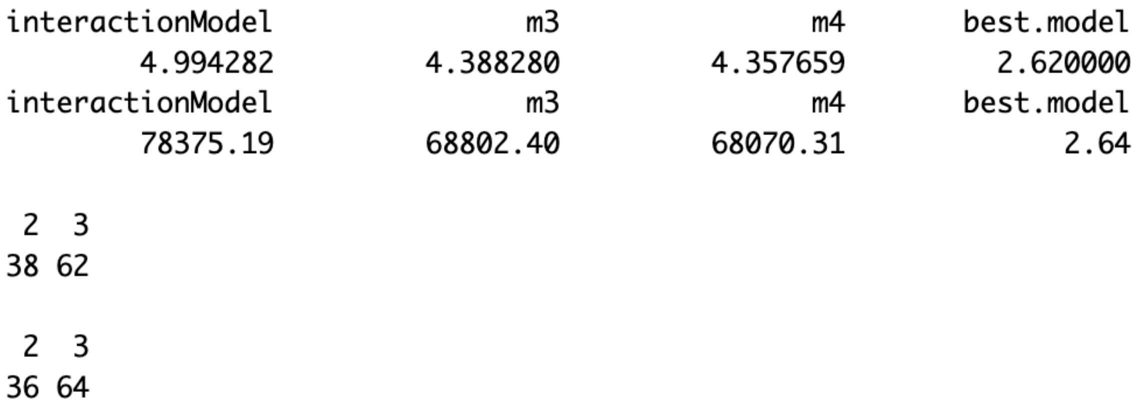
- Predict both systolic blood pressure and log-transformed systolic blood pressure for all observations in the validation sets, and save the residual sums of squares of the validation set in each iteration (for the 2 versions of the response: log and untransformed).

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)
```

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

colMeans(t.rss)
colMeans(rss)

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



```
interactionModel               m3               m4       best.model
        4.994282         4.388280         4.357659         2.620000
interactionModel               m3               m4       best.model
        78375.19         68802.40         68070.31             2.64

 2  3
38 62

 2  3
36 64
```

The third choice of model (`model4`) performs the best most of the time in both predicting the log-transformed systolic blood pressure and the untransformed systolic blood pressure. We can see this based on the low mean RSS of model 3 (compared to that of other models) and the fact that model 3 has the lowest mean RSS the most number of times for predicting both log-transformed and untransformed systolic bp. This result is consistent with what I found in problem 2.

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.

```r
# 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(2,10,25), seed = NA)

# you simulation code here
library(caret)
nsims=200

data <- data.frame(Y = as.numeric(),
          Z1 = as.numeric(),
          Z2 = as.numeric(),
          Z3 = as.numeric(),
          X1 = as.numeric(),
          X2 = as.numeric(),
          X3 = as.numeric(),
          X4 = as.numeric(),
          X5 = as.numeric(),
          X6 = as.numeric(),
          X7 = as.numeric(),
          X8 = as.numeric(),
          X9 = as.numeric(),
          X10 = as.numeric())

best.model <- rep(NA, nsims)

findBest <- function(n, k){

  for(i in 1:nsims){

    # generate predictors
    for(j in 1:n){
      data[j,2:14] <- rnorm(13, 0, 1)
      data[j,"Y"] <- rnorm(1, 1*data[j,"Z1"] + 3*data[j,"Z2"] + 9*data[j,"Z3"], 5)
    }

    # k fold cv
    cv  <- trainControl(method="cv", number=k)
    lm1 <- train(Y ~ Z3,          data=data, method="lm", trControl=cv)
    lm2 <- train(Y ~ Z2 + Z3,     data=data, method="lm", trControl=cv)
    lm3 <- train(Y ~ Z1 + Z2 + Z3, data=data, method="lm", trControl=cv)
    lm4 <- train(Y ~ .,           data=data, method="lm", trControl=cv)

    # get rmse
    rmse <- c(lm1$results$RMSE,
              lm2$results$RMSE,
              lm3$results$RMSE,
              lm4$results$RMSE)

    best.model[i] <- which.min(rmse)
  }

  return(best.model)
}
```

(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`.

We see that across the 3 values of k, model 3 is the best the most number of times, closely followed by model 2. As k becomes larger, model 3 outperforms model 2 more frequently. Models 1 and 4 perform the best the least number of times.

```
k2  <- findBest(n=50, k=2)
k10 <- findBest(n=50, k=10)
k25 <- findBest(n=50, k=25)

best50 <- data.frame("k2"  = c(sum(k2==1),  sum(k2==2),  sum(k2==3),  sum(k2==4)),
                     "k10" = c(sum(k10==1), sum(k10==2), sum(k10==3), sum(k10==4)),
                     "k25" = c(sum(k25==1), sum(k25==2), sum(k25==3), sum(k25==4)))
rownames(best50) <- c("model1", "model2", "model3", "model4")
best50
```

```
##         k2 k10 k25
## model1   8   1   3
## model2 104  94  92
## model3  87  98  99
## model4   1   7   6
```

(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.

The clear winner is $k = 25$, which gives us the most accurate result; in this scenario, model 3 (the correct model) performs the best most often, compared to other scenarios of different k values. $k = 2$ chooses the correct model 48% of the time, $k = 10$ chooses the correct model 47% of the time, and $k = 25$ chooses the correct model 53.5% of the time. Since we often get simpler models than more complex ones, we tend to underfit than overfit the data.

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

```
k2c  <- findBest(n=500, k=2)
k10c <- findBest(n=500, k=10)
k25c <- findBest(n=500, k=25)

best500 <- data.frame("k2"  = c(sum(k2c==1),  sum(k2c==2),  sum(k2c==3),  sum(k2c==4)),
                      "k10" = c(sum(k10c==1), sum(k10c==2), sum(k10c==3), sum(k10c==4)),
                      "k25" = c(sum(k25c==1), sum(k25c==2), sum(k25c==3), sum(k25c==4)))
rownames(best500) <- c("model1", "model2", "model3", "model4")
best500
```

```
##         k2 k10 k25
## model1   0   0   0
## model2  14   3   2
## model3 164 192 187
## model4  22   5  11
```

(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.

There isn't a clear winner for the choice of k when $n = 500$; I think that both $k = 10$ and $k = 25$ give us similar results (although the selection method doesn't perform as well at $k = 2$). When sample size is small, though, we get more underfit models a lot more often. In fact, when sample size is 50, model 2 and model 3 perform the best with similar frequencies, but model 3 performs the best most often by a wide margin when the sample size is 500.

(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!!!).

We can vary the simulation a lot to study how different factors might impact the model outcome. For one, we can have the true predictors, useless variables, and response variable take on a different distribution other than Normal and see whether cross-validation tends to gives us overfit or underfit model, whether it can detect the true distribution of each predictor, what the optimal number of folds is. It can also be a simple change like keeping all the predictors as normally distributed but varying the variances so that they are not all standard normal. Another change we can make is to vary the number of true predictors vs useless variables: like in lecture, we can have different scenarios where we have a few useful predictors and lots of useless variables; many weak predictors; or no useful predictors. Moreover, we can also utilize LASSO and Ridge regression to shrink the coefficient estimates along with doing k-fold cross validations to find the useful predictors and appropriate models. Lastly, we can vary the types of model used; we can use polynomials or other fancier models instead of simple first-order predictors like in this problem.