Solution to multiple regression example

Your name

Install packages

```r
library(MASS)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(glmnet)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

Question 1

```
data1 <- read.csv('math.csv', sep = ';')
data1 <- subset(data1, select = - c(G1,G2))
set.seed(123)
training.samples <- data1$G3 %>% createDataPartition(p = 0.75, list = FALSE) # caret pkg
train.data  <- data1[training.samples, ]
test.data <- data1[-training.samples, ]
```

Question 2

```
fit <- lm(G3~., data = train.data)
fit_step <- stepAIC(fit, k = log(nrow(train.data)), trace = 0) # MASS pkg
pred <- fit_step %>% predict(test.data)
data.frame(
RMSE = RMSE(pred, test.data$G3),
Rsquare = R2(pred, test.data$G3)
)
```

```
##      RMSE  Rsquare
## 1 4.32272 0.147046
```

Question 3

```
library(leaps)
fit_bs <- regsubsets(G3~., data = train.data, nvmax = 30) # leaps pkg
result <- summary(fit_bs)
which.min(result$rss)
```

```
## [1] 30
```

```
result$which[30,]
```

```
##      (Intercept)          schoolMS              sexM               age
##             TRUE              TRUE              TRUE              TRUE
##         addressU        famsizeLE3          PstatusT              Medu
##             TRUE              TRUE             FALSE              TRUE
##             Fedu        Mjobhealth         Mjobother      Mjobservices
##             TRUE              TRUE             FALSE              TRUE
##      Mjobteacher        Fjobhealth         Fjobother      Fjobservices
##             TRUE             FALSE              TRUE              TRUE
##      Fjobteacher        reasonhome       reasonother reasonreputation
##            FALSE              TRUE              TRUE             FALSE
##   guardianmother     guardianother        traveltime         studytime
##            FALSE             FALSE             FALSE              TRUE
```

```
##         failures     schoolsupyes        famsupyes          paidyes
##             TRUE             TRUE             TRUE             TRUE
##    activitiesyes       nurseryyes        higheryes      internetyes
##            FALSE             TRUE             TRUE             TRUE
##      romanticyes           famrel         freetime            goout
##             TRUE             TRUE             TRUE             TRUE
##             Dalc             Walc           health         absences
##             TRUE             TRUE             TRUE             TRUE
```

```
fit_bs <- lm(G3~., data = train.data)
pred <- fit_bs %>% predict(test.data)
data.frame(
RMSE = RMSE(pred, test.data$G3),
Rsquare = R2(pred, test.data$G3)
)
```

```
##       RMSE   Rsquare
## 1 4.233543 0.1933706
```

Question 4

Two models with different predictors, three predictors each

BICs: Smaller -> better

```
mod_step <- lm(G3~Medu +  failures + romantic, data = train.data)
mod_bs <- lm(G3~., data = train.data)
BIC(mod_step) # better
```

```
## [1] 1723.767
```

```
BIC(mod_bs)
```

```
## [1] 1885.456
```

```
pred_step = predict(mod_step, test.data)
pred_bs = predict(mod_bs, test.data)
rmse_step = sqrt(mean((pred_step - test.data$G3)^2))
rmse_bs = sqrt(mean((pred_bs - test.data$G3)^2))
print(paste("step RMSE:", rmse_step))
```

```
## [1] "step RMSE: 4.32272045592812"
```

```
print(paste("best subset RMSE:", rmse_bs))
```

```
## [1] "best subset RMSE: 4.23354261531003"
```

```r
r_squared_step = cor(test.data$G3, pred_step)^2
r_squared_bs = cor(test.data$G3, pred_bs)^2
print(paste("step R^2:", r_squared_step))
```

```
## [1] "step R^2: 0.147046029507003"
```

```r
print(paste("best subset R^2:", r_squared_bs))
```

```
## [1] "best subset R^2: 0.193370636849062"
```

Based on the BIC values of the fitted model using the training data - the one selected by the Stepwise regression using the BIC criterion has a smaller BIC value. This is because the Stepwise procedure uses the BIC critrion while the Best Subset procedure here uses the SSE (RSS) as selection criterion.

According to the RMSE and R2 of the predition for the test data, the Best Subset one is better since R2 always increases (or stays the same) when adding more predictors, this method selected all 30 predictors, even if some of them contribute little predictive power or cause overfitting. The lower RMSE on the testing set suggests that the additional predictors helped capture more variance, but this might be due to chance (overfitting risk).

modifications: 1. Instead of using a single test set RMSE, perform k-fold cross-validation to assess generalizability. This helps to determine whether the extra predictors genuinely improve performance or are just noise. 2. use adjusted R2 3. try to do model diagnosis do see if transformation is needed.