Homework 07 Multiple Linear Regression

Due by 11:59pm, Saturday, March 29, 2025

S&DS 230/530/ENV 757

1) Model Overfitting and R-squared *(20 pts)*

In class, I discussed how, despite its merits, R-squared is not useful as the sole measure of predictive accuracy. The problem is that as more predictors are added (even useless ones), R-squared will always increase (assuming there is no missing data).

This problem is magnified in the case of **Model Overfitting**. We say that a model is over-fit if the number of predictors approaches the number of observations. When the number of predictors is one less than the number of observations, the R-squared will always be 1.

We do a simulation to see this in action.

First, I'll simulate all of my variables (predictors and response) from a normal(0, 1) distribution. That is, the y values and ALL of the possible X predictor values are all randomly chosen from a random normal distribution with mean zero and sd = 1. This means that the x variables are uncorrelated AND they are all NOT significant predictors of Y.

```
set.seed(1)
simdata <- rnorm(10 * 15) # need 10 * 15 values simulated
simdata <- matrix(simdata, nrow = 10, ncol = 15) # now convert this vector into a matrix
colnames(simdata) <- c("y", paste0("x", 1:14)) # add column names
simdata <- as.data.frame(simdata) # convert the matrix into a data frame
```

Let's pretend that this is an actual dataset we'd like to use for linear regression, 10 observations (rows) of 14 predictors and a response variable. The columns are named as follows (so we treat the first column "y" as our response variable and the subsequent columns "x1", "x2", ... as our predictors):

```
colnames(simdata)
```

```
##  [1] "y"   "x1"  "x2"  "x3"  "x4"  "x5"  "x6"  "x7"  "x8"  "x9"  "x10" "x11"
## [13] "x12" "x13" "x14"
```

Note: By virtue of how we simulated the data (independently and identically sampled from a standard normal distribution), "y" has no relationship with any of the 14 possible predictors.

1.1) *(5 pts)* Fit a simple linear regression model predicting `y` using `x1` and save the results to an object called `mod1`. Does it appear that `x1` is a significant predictor of `y`? What is the value of R-squared (use `summary(mod1)$r.squared`)? Interpret its value in the context of the model.

```
mod1 <- lm(y ~ x1, data = simdata)
mod1
```

```
##
## Call:
## lm(formula = y ~ x1, data = simdata)
##
## Coefficients:
## (Intercept)           x1
##      0.2006      -0.2749
```

```
summary(mod1)$r.squared
```

```
## [1] 0.1419054
```

Since the r squared value is 0.142, it appears that x1 is not a significant predictor of y. 14.2% of the variation in y can be explained by the variation in x1.

1.2) *(8 pts)* Using a for-loop, now expand your linear regression model in (a) by iteratively adding in one more predictor at a time. That is to say, the first iteration of your for-loop should use `x1` as a predictor; the second iteration of your for-loop should use `x1`, and `x2`; and so on, until all 14 predictors are used in your model. Store the values of R-squared in a vector called `rsqvals` of length 14, so that the `rsqvals[i]` should contain the value of R-squared for a model using predictors `x1` through `xi`. Finally, display the values in `rsqvals`.

Hint: remember the shorthand formula to include all predictors: `y ~ .`. So, for example, if I wanted to fit a model using `x1` through `x7` as predictors, I could do:

```
simtemp <- simdata[,1:8]
m7 <- lm(y ~ ., data = simtemp)
```

```
rsqvals <- c()
for (x in 1:14) {
  simtemp <- simdata[,1:(x+1)]
  # print(simtemp)
  tempmodel <- lm(y ~ ., simtemp)
  rsqvals[x] = summary(tempmodel)$r.squared
}
rsqvals
```

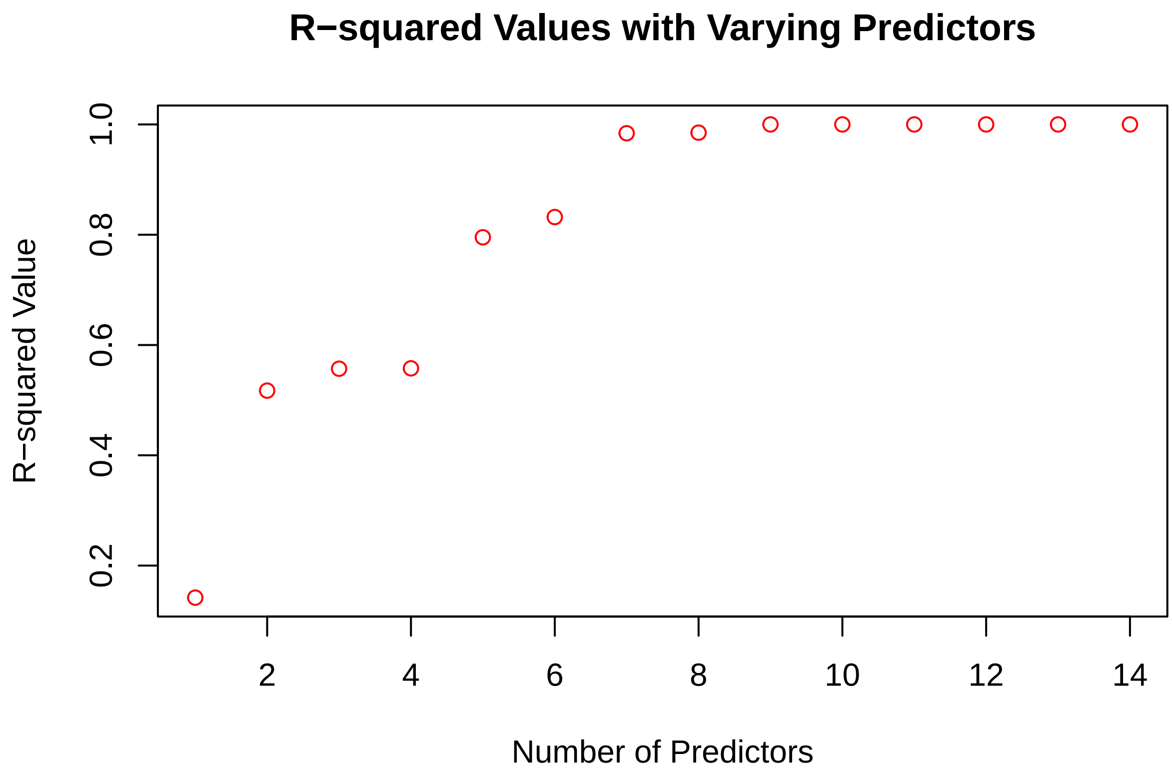
```
##  [1] 0.1419054 0.5173063 0.5570001 0.5577011 0.7953346 0.8320571 0.9840191
##  [8] 0.9851084 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
```

1.3 *(7 pts)* How many predictors did it take to reach an R-squared of 1 (write a line of code to get this value)? Display a plot that shows the increase in R-squared with an increasing number of predictors. Be sure to label your plot.

```
match(1, rsqvals)
```

```
## [1] 9
```

```
plot(rsqvals,
     main="R-squared Values with Varying Predictors",
     xlab="Number of Predictors",
     ylab="R-squared Value",
     col='red')
```



It took 9 predictors, which makes sense since with 9 slopes and 1 intercept we have 10 variables, and a linear system of 10 variables and 10 linearly independent equations always has a unique solution.

2) Ohio Crime Data *(80 points)*

A 1999 survey sponsored by the US Justice Department and the University of Cincinnatti interviewed a number of Ohio residents on their attitudes toward crime, criminals, and ways of reducing crime. In addition, various religious and demographic information was collected.

A summary of survey questions (and label information) can be found HERE. You'll want to look at pages 17 - 34 at a minimum. Each question is labeled V1, V2, etc. through V98. We'll be looking at a subset of these questions.

The data itself is HERE.

2.1) *(2 pts)* Read the data into an object called `crime`. Get the dimension and column names of `crime`. You won't need the `as.is = TRUE` option.

```
crime <- read.csv('https://raw.githubusercontent.com/jreuning/sds230_data/refs/heads/main/ohiocrime.csv
dim(crime)
```

```
## [1] 559  99
```

```
names(crime)
```

```
##  [1] "V1"     "V2"     "V3"     "V4"     "V5"     "V6"     "V7"     "V8"
##  [9] "V9"     "V10"    "V11"    "V12"    "V13"    "V14"    "V15"    "V16"
## [17] "V17"    "V18"    "V19"    "V20"    "V21"    "V22"    "V23"    "V24"
```

```
## [25] "V25"    "V26"    "V27"    "V28"    "V29"    "V30"    "V31"    "V32"   
## [33] "V33"    "V34"    "V35"    "V36"    "V37"    "V38"    "V39"    "V40"   
## [41] "V41"    "V42"    "V43"    "V44"    "V45"    "V46"    "V47"    "V48"   
## [49] "V49"    "V50"    "V51"    "V52"    "V53"    "V54"    "V55"    "V56"   
## [57] "V57"    "V58"    "V59"    "V60"    "V61"    "V62"    "V63"    "V64"   
## [65] "V65"    "V66"    "V67"    "V68"    "V69"    "V70"    "V71"    "V72"   
## [73] "V73"    "V74"    "V75"    "V76"    "V77"    "V78"    "V79"    "V80"   
## [81] "V81"    "V82"    "V83"    "V84"    "V85"    "V86"    "V87"    "V88"   
## [89] "V89"    "V90"    "V91"    "V92"    "V93"    "V94"    "V95"    "V96"   
## [97] "V97"    "V98"    "CASENO"
```

```r
head(crime)
```

```
##   V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21
## 1  0 21  1 19  0  1 10  3  1   4   3   4   5   6   3   4   6   5   4   6   5
## 2  1 22  1 14  2  0  0  3  1   5   1   1   6   6   1   5   3   4   5   5   4
## 3  1 32  0 13  2  1 24  3  4   6   1   6   6   6   1   6   1   1   6   6   6
## 4  1 23  1 11  1  1 20  3  1   2   6   1   6   6   1   1   6   4   1   4   6
## 5  1 21  0  7  0  1 13  2  2   2   5   1   5   6   2   1   6   6   5   6   4
## 6  1 27  1 19  1  2  0  2  3   4   6   2   6   6   3   3   5   5   4   6   6
##   V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38 V39 V40
## 1   4   5   5   4   3   6   5   6   4   3   4   3   1   1   3   1   3   1   4
## 2   5   3   6   6   4   5   4   5   3   3   3   3   1   2   2   1   2   2   2
## 3   1   6   5   6   1   6   6   6   1   2   2   1   1   2   1   1   2   2   6
## 4   2   1   2   2   1   2   2   6   3   5   3   3   1   2   3   1   2   1   2
## 5   5   5   2   2   2   2   2   5   5   4   5   3   1   1   2   1   1   1   4
## 6   5   4   6   5   3   5   4   5   2   2   2   2   1   3   2   1   1   1   5
##   V41 V42 V43 V44 V45 V46 V47 V48 V49 V50 V51 V52 V53 V54 V55 V56 V57 V58 V59
## 1   5   3   5   4   2   3   4   1   3   2   5   4   6   6   5   4   4   5   4
## 2   5   3   4   4   2   2   4   4   4   4   4   4   4   5   4   5   5   5   4
## 3   6   6   4   6   4   2   6   2   6   2   6   6   6   6   6   1   1   1   1
## 4   6   2   5   3   4   1   2   2   5   1   4   2   5   6   4   2   6   5   4
## 5   5   3   4   5   5   4   5   4   5   2   6   4   2   4   6   5   2   4   4
## 6   5   5   5   5   5   5   5   5   4   3   5   5   4   5   5   4   4   5   3
##   V60 V61 V62 V63 V64 V65 V66 V67 V68 V69 V70 V71 V72 V73 V74 V75 V76 V77 V78
## 1   5   2   5   5   2   4   5   4   4  69   1   1   5   2   2   2   2   2   1
## 2   4   4   4   4   2   4   5   4   5  23   1   1   5   2   2   2   2   2   2
## 3   1   1   1   4   6   1   1   1   2  64   0   1   2   2   2   1   1   1   1
## 4   3   6   5   6   4   5   2   5   2  55   1   1   7   2   1   2   1   2   1
## 5   4   2   4   4   2   4   4   4   4  50   1   1   6   2   2   2   2   2   2
## 6   4   3   5   5   3   2   4   5   3  27   0   1   7   2   2   2   2   2   1
##   V79 V80 V81 V82 V83 V84 V85 V86 V87 V88        V89 V90 V91 V92 V93 V94 V95 V96
## 1   2   2   2   1   2   2   2   7   5   3                1   2   1   2   3   3   3
## 2   2   2   2   2   2   2   2   5   2   2                2   1   1   1   3   3   3
## 3   2   2   1   1   2   2   3   5   1   5                1   1   1   1   3   1   3
## 4   2   2   2   2   2   2   1   8   6   1  lutheran      2   2   2   1   2   2   3
## 5   2   2   2   2   2   2   1   7   5   1                1   1   1   1   3   3   3
## 6   2   2   2   2   2   2   2   8   1   1 episcopal      2   1   1   1   3   3   3
##   V97 V98 CASENO
## 1   3   3      1
## 2   3   3      2
## 3   2   1      3
## 4   3   3      4
## 5   3   3      5
```

```
## 6   3   3      6
```

2.2) *(10 pts)* First consider the variables in columns 10 through 23; these are 14 questions having to do with attitudes toward preventing crime (see PDF file). Each question is on a 6 point scale (see PDF file for particular levels). Your first task is to visually examine the correlations with the `corrplot.mixed` function discussed in class 11.

I've given an outline chunk below. Your job is to fill in the details as indicated. Be sure to remove `eval = F` before knitting.

```r
#note the options above are to make plots work properly in the corrplot package.

#Load the corrplot package
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.4.3

## corrplot 0.95 loaded
```

```r
#calculate pairwise correlations for columns 10-23 of crime.  You'll need the   use = "pairwise.complet
cor1 <- cor(crime[10:23], use="pairwise.complete.obs", method="pearson")

#round cor1 to 2 decimal places and display the result.
cor1 <- round(cor1, digits=2)
cor1
```

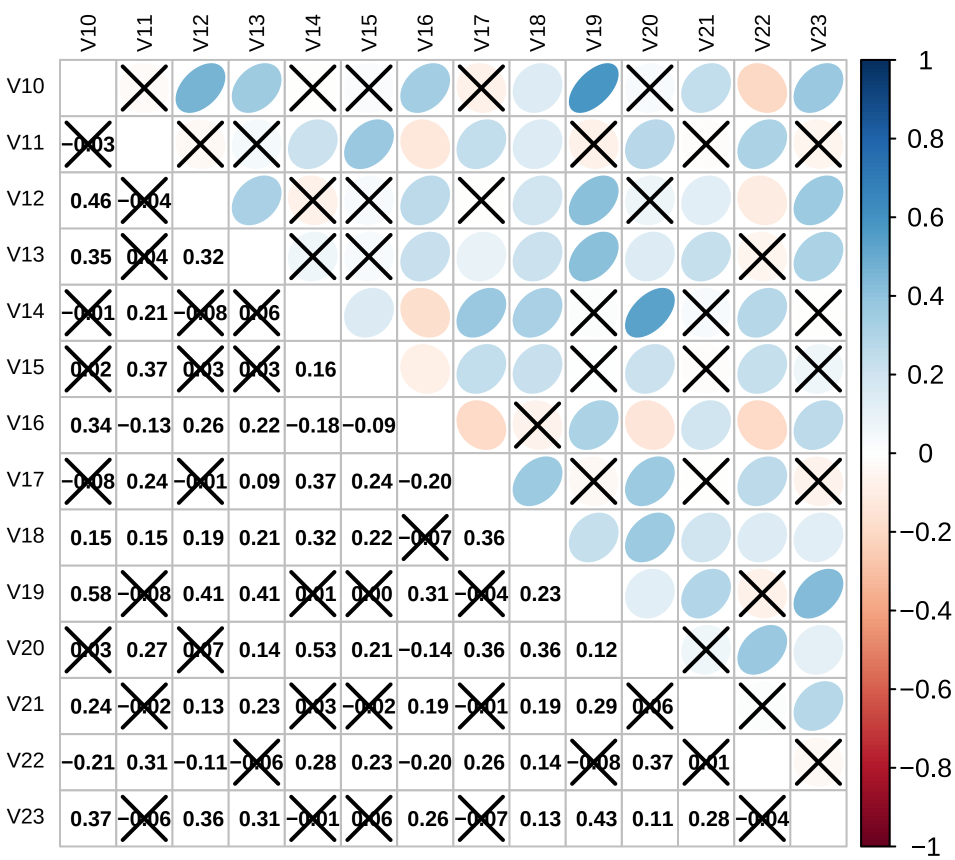
```
##       V10   V11   V12   V13   V14   V15   V16   V17   V18   V19   V20   V21
## V10  1.00 -0.03  0.46  0.35 -0.01  0.02  0.34 -0.08  0.15  0.58  0.03  0.24
## V11 -0.03  1.00 -0.04  0.04  0.21  0.37 -0.13  0.24  0.15 -0.08  0.27 -0.02
## V12  0.46 -0.04  1.00  0.32 -0.08  0.03  0.26 -0.01  0.19  0.41  0.07  0.13
## V13  0.35  0.04  0.32  1.00  0.06  0.03  0.22  0.09  0.21  0.41  0.14  0.23
## V14 -0.01  0.21 -0.08  0.06  1.00  0.16 -0.18  0.37  0.32  0.01  0.53  0.03
## V15  0.02  0.37  0.03  0.03  0.16  1.00 -0.09  0.24  0.22  0.00  0.21 -0.02
## V16  0.34 -0.13  0.26  0.22 -0.18 -0.09  1.00 -0.20 -0.07  0.31 -0.14  0.19
## V17 -0.08  0.24 -0.01  0.09  0.37  0.24 -0.20  1.00  0.36 -0.04  0.36 -0.01
## V18  0.15  0.15  0.19  0.21  0.32  0.22 -0.07  0.36  1.00  0.23  0.36  0.19
## V19  0.58 -0.08  0.41  0.41  0.01  0.00  0.31 -0.04  0.23  1.00  0.12  0.29
## V20  0.03  0.27  0.07  0.14  0.53  0.21 -0.14  0.36  0.36  0.12  1.00  0.06
## V21  0.24 -0.02  0.13  0.23  0.03 -0.02  0.19 -0.01  0.19  0.29  0.06  1.00
## V22 -0.21  0.31 -0.11 -0.06  0.28  0.23 -0.20  0.26  0.14 -0.08  0.37  0.01
## V23  0.37 -0.06  0.36  0.31 -0.01  0.06  0.26 -0.07  0.13  0.43  0.11  0.28
##       V22   V23
## V10 -0.21  0.37
## V11  0.31 -0.06
## V12 -0.11  0.36
## V13 -0.06  0.31
## V14  0.28 -0.01
## V15  0.23  0.06
## V16 -0.20  0.26
## V17  0.26 -0.07
## V18  0.14  0.13
## V19 -0.08  0.43
## V20  0.37  0.11
```

```
## V21  0.01  0.28
## V22  1.00 -0.04
## V23 -0.04  1.00
```

```r
#finds the exact cell of cor1 which contains the maximum positive pairwise
# correlation (other than 1), and stores that cell in maxloc
maxloc <- which(cor1 == max(cor1[cor1<1]), arr.ind = TRUE)

#prints the names of the columns of maxloc, which would be the two columns
# with the maximum positive pairwise correlation
names(crime[10:23])[maxloc[1,]]
```

```
## [1] "V19" "V10"
```

```r
#Create an object called sigcorr that has the results of cor.mtest for columns 10-23 of the crime data.
sigcorr <- cor.mtest(crime[10:23], conf.level = 0.95)

#Use corrplot.mixed to display confidence ellipses, pairwise correlation values, and put on 'X' over no
corrplot.mixed(cor1, lower.col="black", upper = "ellipse", tl.col = "black", number.cex=.7,
               tl.pos = "lt", tl.cex=.7, p.mat = sigcorr$p, sig.level = .05)
```

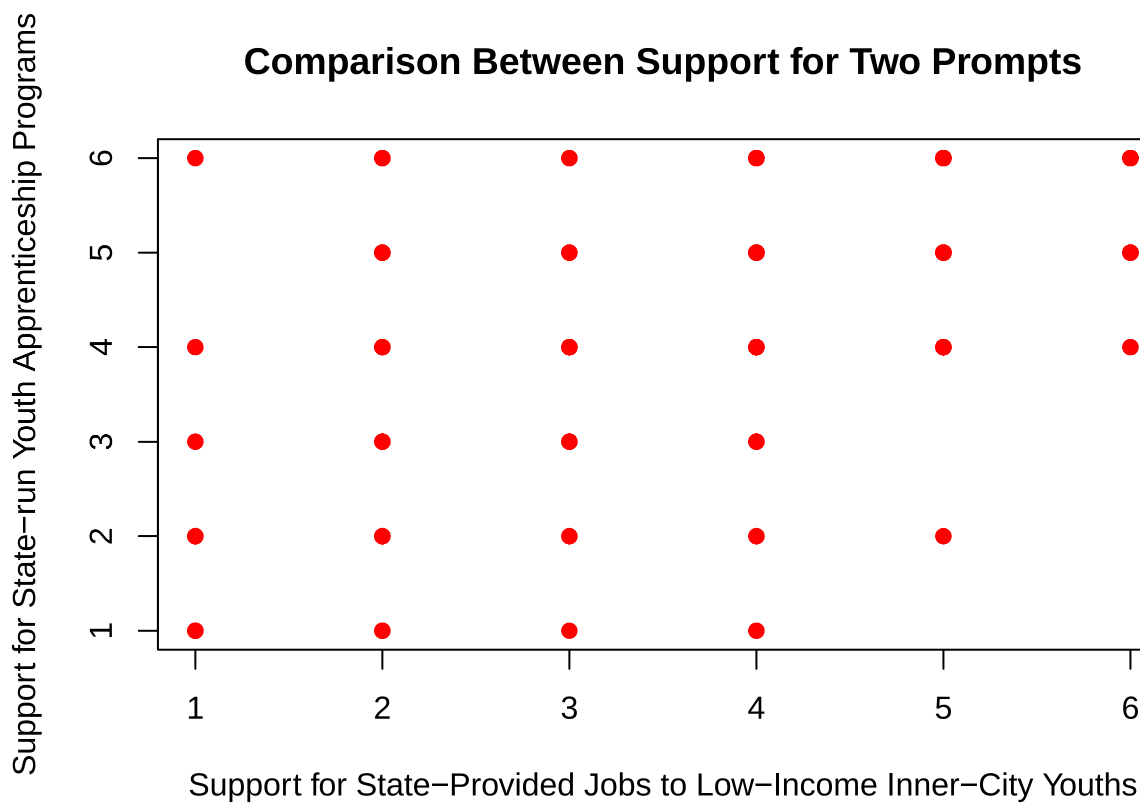


2.3) *(5 pts)* Comment on the overall level of correlations among the considered questions. Which pair of questions had the highest sample correlation? Are you surprised? (include comments on the actual questions in your answer)

Questions 10 and 19 had the highest pairwise sample correlation; question 10 was about state-provided jobs to low-income inner-city youths, and question 19 was similar, asking about state-provided apprenticeship programs for youths in general. I'm not surprised that these two questions had a high correlation given that they ask very similar questions, the difference being that question 10 is about giving jobs directly rather than apprenticeship programs, and that question 10 specifies low-income inner-city youths.

2.4) *(5 pts)* Make a scatterplot of values for the two questions that had the highest pairwise correlation. Make sure your plot has labels for each axis (and not 'V10' - something with meaning). Include two top titles - one for the plot as a whole, one which reports the sample correlation to two decimal places. How helpful is this plot?

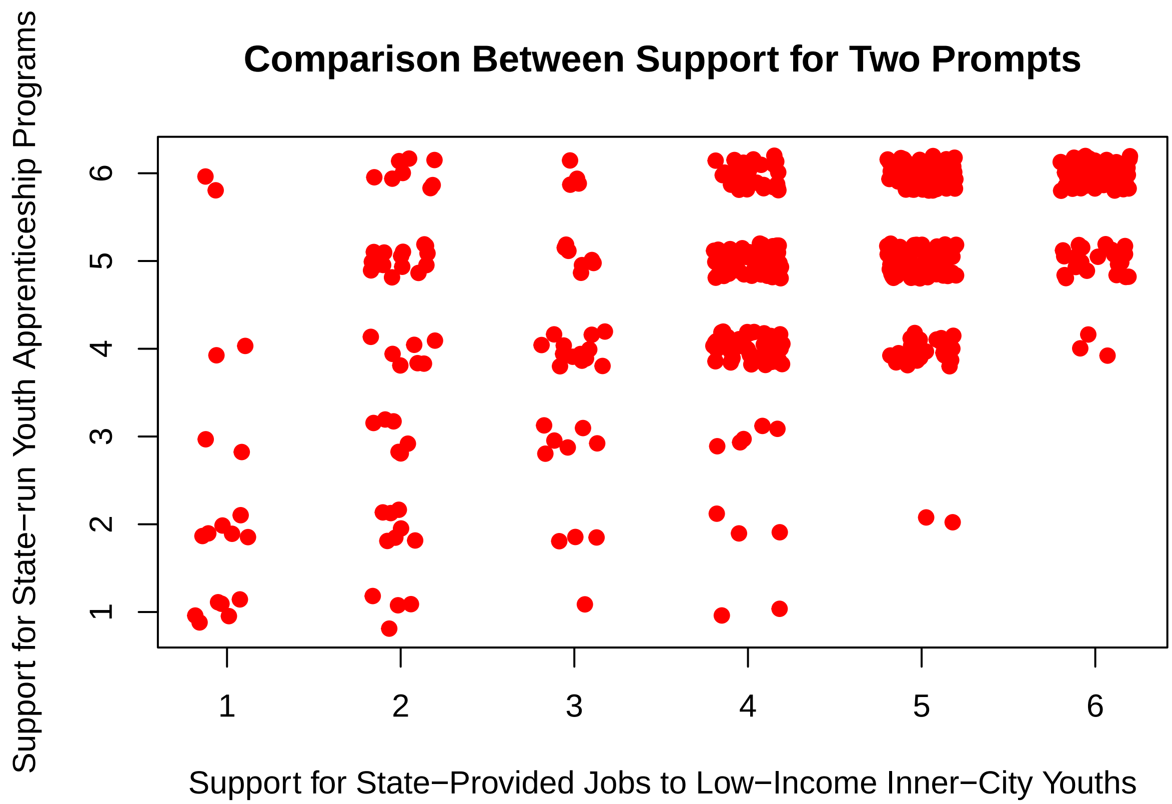
```
plot(x=crime$V10,
     y=crime$V19,
     col="red",
     main="Comparison Between Support for Two Prompts",
     xlab="Support for State-Provided Jobs to Low-Income Inner-City Youths",
     ylab="Support for State-run Youth Apprenticeship Programs",
     pch=19)
```



This plot is not helpful since there are a discrete number of possible response pairs, and there are multiple counts for each possible response pair which cannot be shown since the points cover each other.

2.5) *(4 pts)* Repeat part 2.4) but jitter results in both directions. Write a sentence about what you observe.

```r
plot(x=jitter(crime$V10, factor=1),
     y=jitter(crime$V19, factor=1),
     col="red",
     main="Comparison Between Support for Two Prompts",
     xlab="Support for State-Provided Jobs to Low-Income Inner-City Youths",
     ylab="Support for State-run Youth Apprenticeship Programs",
     pch=19)
```



There appears to be a positive correlation, in that there are a lot of points for when the support for the two questions are similar.

We are now going to proceed with performing stepwise regression. In particular, we're going to fit a model that looks at possible predictors of question V45 (you'll want to look up what this question is). To do this, I'm making a new dataset called `crime2` which contains the relevant columns (notice I'm putting the response variable FIRST). Be sure to remove the option `eval = F`.

```r
crime2 <- crime[, c(45, 10:23, 65, 70, 72, 87, 86)]
names(crime2)
```

```
##  [1] "V45" "V10" "V11" "V12" "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20"
## [13] "V21" "V22" "V23" "V65" "V70" "V72" "V87" "V86"
```

```r
dim(crime2)
```

```
## [1] 559  20
```

2.6) *(4 pts)* Perform best subsets regression using the `regsubsets` function in the `leaps` package. Save the results in an object called `mod2`. Get the summary of `mod2` and save the results in an object called `mod2sum`. Display `mod2sum$which` to get a sense of which variables are included at each step of best subsets.

```r
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.4.3
```

```r
mod2 <- regsubsets(V45 ~ ., data = crime2, nvmax = 19)
mod2sum <- summary(mod2)
mod2sum$which
```

```
##    (Intercept)   V10   V11   V12   V13   V14   V15   V16   V17   V18   V19
## 1         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## 3         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## 4         TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## 5         TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## 6         TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE
## 7         TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE
## 8         TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE
## 9         TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## 10        TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## 11        TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## 12        TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## 13        TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## 14        TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 15        TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 16        TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 17        TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 18        TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 19        TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##      V20   V21   V22   V23   V65   V70   V72   V87   V86
## 1  FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 2  FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 3  FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
## 4   TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## 5   TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
## 6   TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
## 7   TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 8   TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 9   TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 10  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 11  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 12  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 13  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 14  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 15  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE
## 16  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
## 17  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## 18  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 19  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

2.7) *(8 pts)* Following the example in classes 13-16, examine the best model according to highest r-squared. Here are your steps:

- Make an object called `modnum` which contains the row number in `mod2sum$which` for the model with the highest r-squared.
- Print the variable names for predictors that ended up in this model.
- Make a temporary dataset called `crimetemp` which has the columns of `crime2` that were included in this model.
- Fit the model and return summary information for the model.

```r
modnum <- which.max(mod2sum$rsq)
modnum
```

```
## [1] 19
```

```r
names(crime2)[mod2sum$which[modnum, ]][-1]
```

```
##  [1] "V10" "V11" "V12" "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20" "V21"
## [13] "V22" "V23" "V65" "V70" "V72" "V87" "V86"
```

```r
crimetemp <- crime2[ ,mod2sum$which[modnum, ]]
summary(lm(V45 ~ ., data = crimetemp))
```

```
##
## Call:
## lm(formula = V45 ~ ., data = crimetemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4473 -0.9458  0.1761  0.9869  3.5086
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.85336    0.72565   3.932 9.70e-05 ***
## V10         -0.15773    0.06449  -2.446  0.01482 *
## V11          0.00311    0.03983   0.078  0.93780
## V12         -0.03197    0.04581  -0.698  0.48558
## V13          0.07778    0.07564   1.028  0.30430
## V14          0.04174    0.08821   0.473  0.63632
## V15          0.06909    0.05011   1.379  0.16858
## V16         -0.07456    0.04844  -1.539  0.12442
## V17          0.10823    0.04769   2.269  0.02370 *
## V18          0.08858    0.06488   1.365  0.17281
## V19         -0.03782    0.07076  -0.535  0.59322
## V20          0.14524    0.07491   1.939  0.05314 .
## V21         -0.04619    0.07627  -0.606  0.54502
## V22          0.12693    0.04612   2.752  0.00615 **
## V23         -0.02685    0.06019  -0.446  0.65572
## V65          0.07866    0.04668   1.685  0.09266 .
## V70         -0.06443    0.14128  -0.456  0.64858
## V72         -0.14732    0.03518  -4.188 3.37e-05 ***
## V87          0.01868    0.04679   0.399  0.68996
```

```
## V86          0.01602    0.04659   0.344  0.73107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.354 on 465 degrees of freedom
##   (74 observations deleted due to missingness)
## Multiple R-squared:  0.254,  Adjusted R-squared:  0.2236
## F-statistic: 8.334 on 19 and 465 DF,  p-value: < 2.2e-16
```

2.8) *(4 pts)* Repeat 2.7) for adjusted R-squared.

```r
modnum <- which.max(mod2sum$adjr2)
modnum
```

```
## [1] 9
```

```r
names(crime2)[mod2sum$which[modnum, ]][-1]
```

```
## [1] "V10" "V15" "V16" "V17" "V18" "V20" "V22" "V65" "V72"
```

```r
crimetemp <- crime2[ ,mod2sum$which[modnum, ]]
summary(lm(V45 ~ ., data = crimetemp))
```

```
##
## Call:
## lm(formula = V45 ~ ., data = crimetemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4910 -0.9754  0.2111  1.0109  3.2550
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.11348    0.48454   6.426 3.05e-10 ***
## V10         -0.18058    0.05109  -3.535 0.000446 ***
## V15          0.06140    0.04540   1.352 0.176841
## V16         -0.10858    0.04380  -2.479 0.013509 *
## V17          0.14537    0.04487   3.239 0.001277 **
## V18          0.07571    0.05953   1.272 0.204049
## V20          0.16818    0.06606   2.546 0.011192 *
## V22          0.12110    0.04198   2.885 0.004087 **
## V65          0.05647    0.04373   1.291 0.197180
## V72         -0.14903    0.03028  -4.922 1.16e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.342 on 503 degrees of freedom
##   (46 observations deleted due to missingness)
## Multiple R-squared:  0.252,  Adjusted R-squared:  0.2386
## F-statistic: 18.82 on 9 and 503 DF,  p-value: < 2.2e-16
```

2.9) *(4 pts)* Repeat 2.7) for BIC.

```r
modnum <- which.min(mod2sum$bic)
modnum
```

```
## [1] 5
```

```r
names(crime2)[mod2sum$which[modnum, ]][-1]
```

```
## [1] "V10" "V17" "V20" "V22" "V72"
```

```r
crimetemp <- crime2[ ,mod2sum$which[modnum, ]]
summary(lm(V45 ~ ., data = crimetemp))
```

```
##
## Call:
## lm(formula = V45 ~ ., data = crimetemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3112 -1.0403  0.2205  1.0045  3.2482
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.01521    0.41236   7.312 9.79e-13 ***
## V10         -0.21185    0.04715  -4.493 8.65e-06 ***
## V17          0.19763    0.04265   4.634 4.52e-06 ***
## V20          0.21932    0.06301   3.481 0.000541 ***
## V22          0.12923    0.04112   3.143 0.001768 **
## V72         -0.14527    0.02926  -4.965 9.29e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.364 on 529 degrees of freedom
##   (24 observations deleted due to missingness)
## Multiple R-squared:  0.2264, Adjusted R-squared:  0.2191
## F-statistic: 30.97 on 5 and 529 DF,  p-value: < 2.2e-16
```

2.10) *(4 pts)* Repeat 2.7) for the Cp Statistic.

```r
modnum <- min(c(1:length(mod2sum$cp))[mod2sum$cp <= c(1:length(mod2sum$cp)) + 1])
modnum
```

```
## [1] 6
```

```r
names(crime2)[mod2sum$which[modnum, ]][-1]
```

```
## [1] "V10" "V16" "V17" "V20" "V22" "V72"
```

```r
crimetemp <- crime2[ ,mod2sum$which[modnum, ]]
summary(lm(V45 ~ ., data = crimetemp))
```

```
##
## Call:
## lm(formula = V45 ~ ., data = crimetemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4814 -1.0093  0.2653  1.0080  3.0635
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.47707    0.44687   7.781 3.82e-14 ***
## V10         -0.17115    0.04971  -3.443 0.000621 ***
## V16         -0.11553    0.04343  -2.660 0.008054 **
## V17          0.18425    0.04267   4.319 1.88e-05 ***
## V20          0.20605    0.06291   3.275 0.001125 **
## V22          0.11971    0.04098   2.921 0.003639 **
## V72         -0.15368    0.02946  -5.216 2.63e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.354 on 527 degrees of freedom
##   (25 observations deleted due to missingness)
## Multiple R-squared:  0.2378, Adjusted R-squared:  0.2291
## F-statistic:  27.4 on 6 and 527 DF,  p-value: < 2.2e-16
```

2.11) *(6 pts)* We choose as our final model the model indicated by BIC. Go refit this model and save the results in an object called `modfin`. How many observations had missing values in this model vs. the number of observations with missing values in the model with all predictors?

```r
modnum <- which.min(mod2sum$bic)
modnum
```

```
## [1] 5
```

```r
names(crime2)[mod2sum$which[modnum, ]][-1]
```

```
## [1] "V10" "V17" "V20" "V22" "V72"
```

```r
crimetemp <- crime2[ ,mod2sum$which[modnum, ]]
modfin <- lm(V45 ~ ., data = crimetemp)
summary(modfin)
```

```
##
## Call:
## lm(formula = V45 ~ ., data = crimetemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3112 -1.0403  0.2205  1.0045  3.2482
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  3.01521    0.41236   7.312 9.79e-13 ***
## V10         -0.21185    0.04715  -4.493 8.65e-06 ***
## V17          0.19763    0.04265   4.634 4.52e-06 ***
## V20          0.21932    0.06301   3.481 0.000541 ***
## V22          0.12923    0.04112   3.143 0.001768 **
## V72         -0.14527    0.02926  -4.965 9.29e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.364 on 529 degrees of freedom
##   (24 observations deleted due to missingness)
## Multiple R-squared:  0.2264, Adjusted R-squared:  0.2191
## F-statistic: 30.97 on 5 and 529 DF,  p-value: < 2.2e-16
```

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

```
## [1] 0.2264285
```

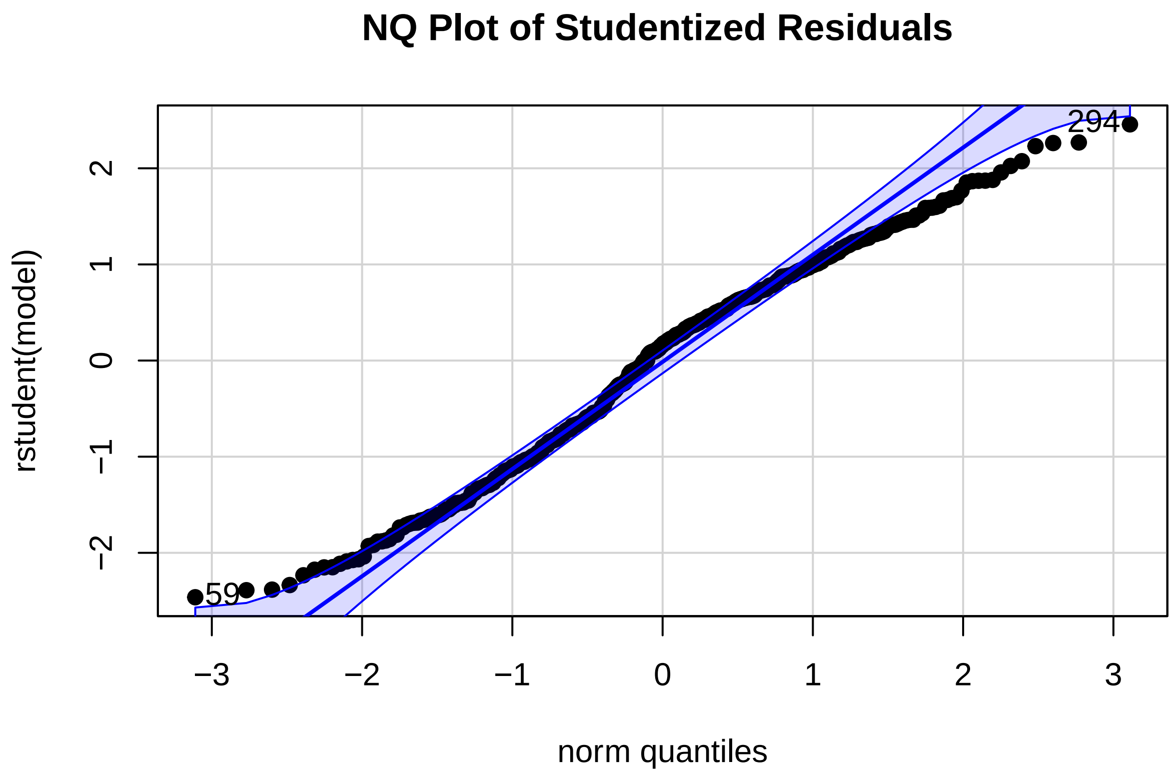
The final model had 24 observations removed due to missing values, as opposed to 74 for the model that contained all variables.

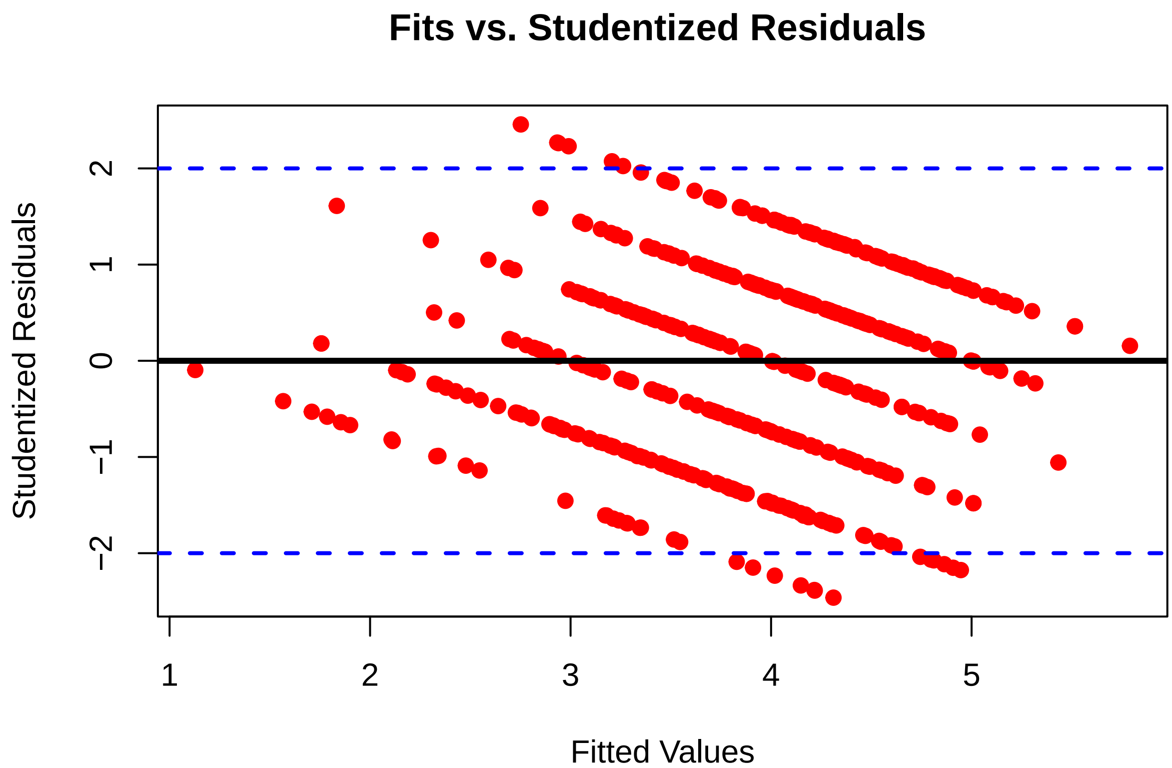
2.12) *(6 pts)* Make two residual plots - studentized residuals vs fitted values (with boundaries at +/-2 and +/-3) as well a normal quantile plot of the residuals. Write a few sentences about how the plots do/do not indicate that we've met the assumptions of our regression model. Note : the six-toed beast that seems to have slashed the plot of fits vs. residuals is exactly what we would expect. Why is this?

```r
library(car)
```

```
## Loading required package: carData
```

```r
myResPlots <- function(model, label){

  #Normal quantile plot of studentized residuals
  qqPlot(rstudent(model), pch = 19, main = paste("NQ Plot of Studentized Residuals", label))

  #plot of fitted vs. studentized residuals
  plot(rstudent(model) ~ model$fitted.values, pch = 19, col = 'red', xlab = "Fitted Values", ylab = "St
     main = paste("Fits vs. Studentized Residuals", label))
  abline(h = 0, lwd = 3)
  abline(h = c(2,-2), lty = 2, lwd = 2, col="blue")
  abline(h = c(3,-3), lty = 2, lwd = 2, col="green")

}

myResPlots(modfin, label = "")
```

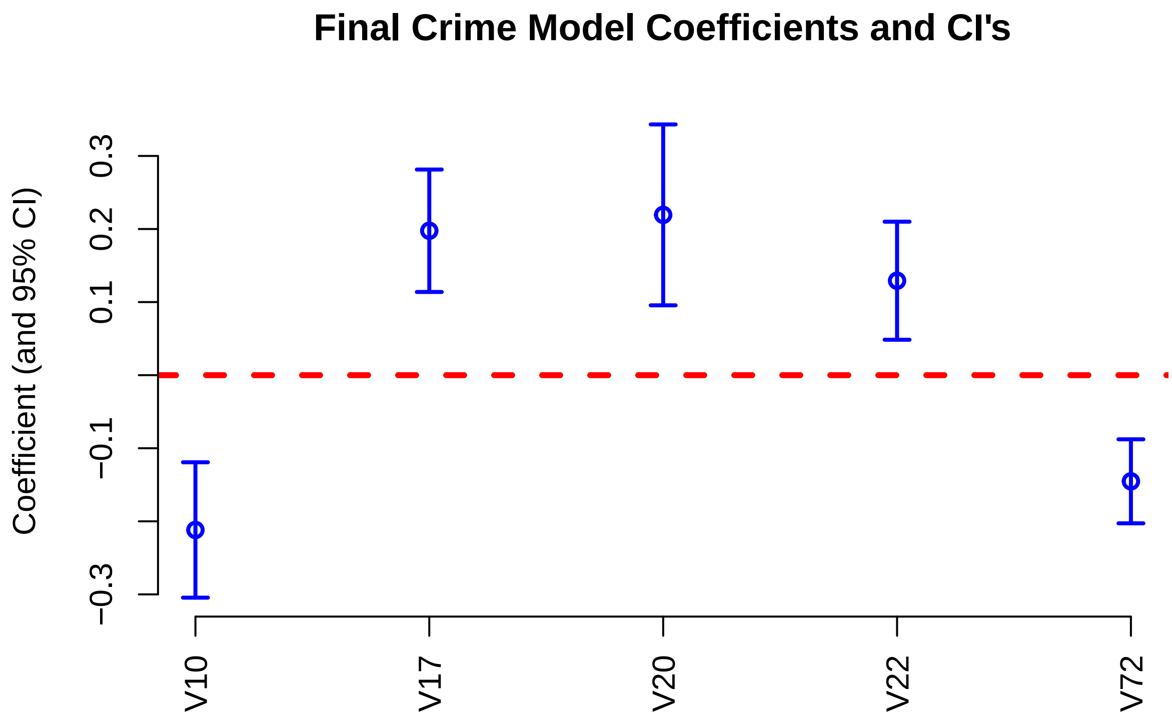

NQ Plot of Studentized Residuals
rstudent(model)
norm quantiles
−3
−2
−1
0
1
2
3
2
1
0
−1
−2
294
59



Looking at the normal quantile plot, the residuals do seem to lie within the confidence interval at least for the values closer to the center, which suggests that the residuals do follow a normal distribution. Then, looking at the studentized residuals vs fitted values, for the most part there doesn't seem to be a significant difference in the residuals between the different possible fitted values, which is good. The reason that the plot looks like 6 straight lines is because for any fitted value, there are only 6 possible residuals due to question 45 only having 6 possible responses. And as the fitted value increases, the residuals decrease by that same amount.

2.13) *(5 pts)* Run the code below. Describe what this does in not more than two sentences.

```
CIs <- confint(modfin)
coefs <- coef(modfin)
library(plotrix)
plotCI(1:(length(coefs)-1), coefs[-1],
       ui = CIs[-1,2], li = CIs[-1,1],
       xlab = "",
       ylab = "Coefficient (and 95% CI)",
       main = "Final Crime Model Coefficients and CI's",
       axes = FALSE, lwd = 2, col = "blue")
abline(h = 0, lty = 2, lwd = 3, col = "red")
axis(side = 2)
axis(side = 1, at = 1:(length(coefs)-1), label = names(coefs)[-1], las = 2)
```



The code calculates and displays the 95% confidence intervals of the correlation coefficients for each of the 5 variables included in the final model.

2.14) *(13 pts)* FINALLY - write a short paragraph discussing your model results.

- Comment on R-squared
- Comment on direction and interpretation of each of the predictors included in the final model. Are you surprised in any instance?

The R-squared value was 0.226, which suggests that 0.226 of the variation in the responses to question 45 is explained by the variation in the responses to the questions of the predictors included in the final model. Questions 10 and 72 were negative predictors for question 45, whereas questions 17, 20, and 22 were positive predictors. The correlation for question 72 is a little surprising; it suggests that those who are more educated are less likely to think vindictively towards criminals - I thought it'd be the other way around since those who are more educated are less likely to commit (violent) crime. The results for questions 17, 20, and 22 are not surprising at all since they all are directly related to harsher punishments for criminals.

THE END