Problem Set 5

QTM 200: Applied Regression Analysis

Due: March 4, 2020

Instructions

- Please show your work! You may lose points by simply writing in the answer. If the problem requires you to execute commands in `R`, please include the code you used to get your answers. Please also include the `.R` file that contains your code. If you are not sure if work needs to be shown for a particular problem, please ask.
- Your homework should be submitted electronically on the course GitHub page in `.pdf` form.
- This problem set is due at the beginning of class on Wednesday, March 4, 2020. No late assignments will be accepted.
- Total available points for this homework is 100.

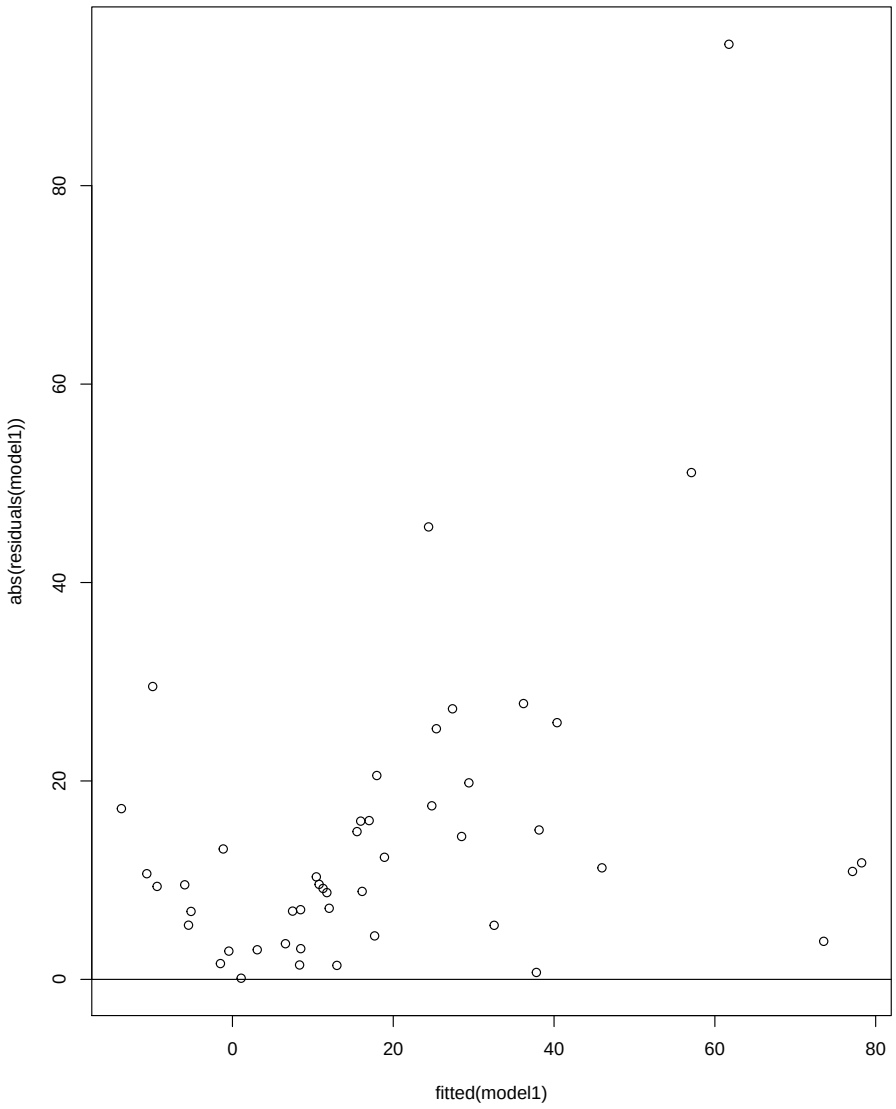
Using the `teengamb` dataset, fit a model with `gamble` as the response and the other variables as predictors.

```
# load data
gamble <- (data=teengamb)
# run regression on gamble with specified predictors
model1 <- lm(gamble ~ sex + status + income + verbal, gamble)

install.packages("car")
library("car")
```

Answer the following questions:

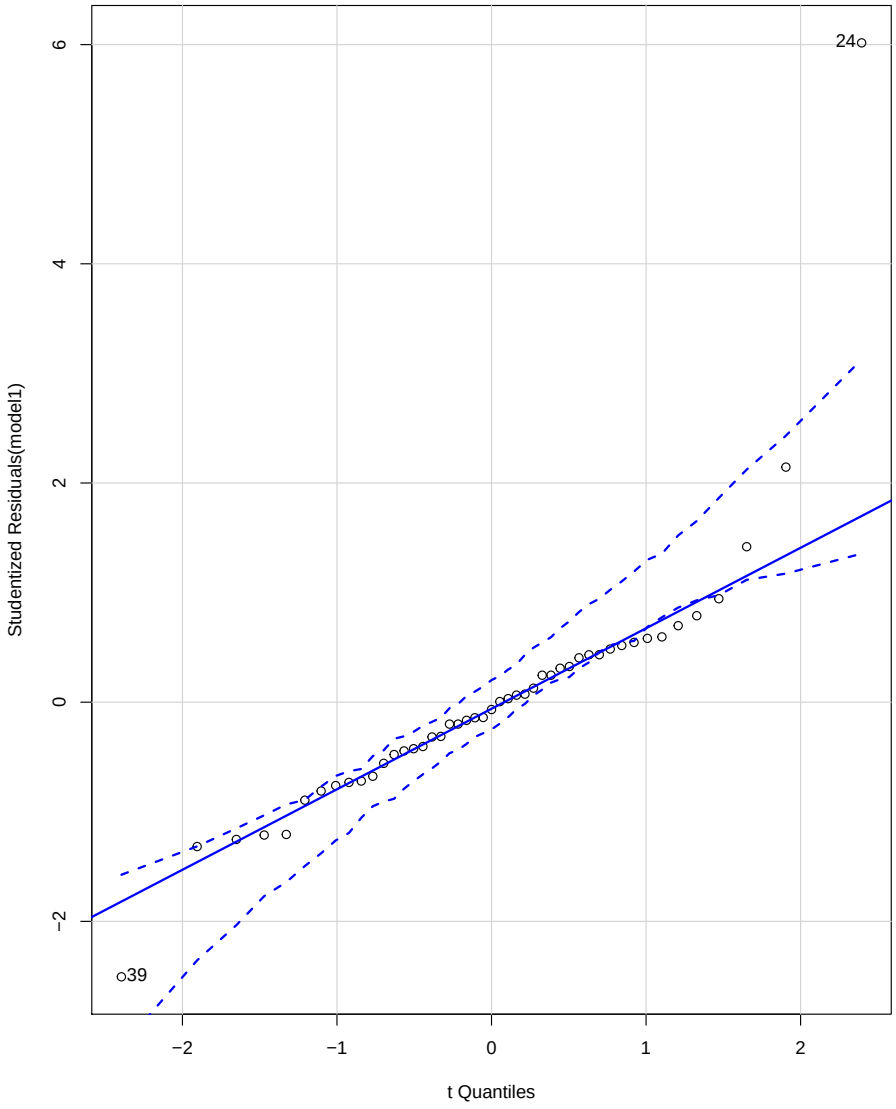
(a) Check the constant variance assumption for the errors by plotting the residuals versus the fitted values.

```r
#a)
plot(residuals(model1) ~ fitted(model1), data=gamble)
abline(h=0)

# Use absolute residuals
plot(abs(residuals(model1)) ~ fitted(model1), data=gamble)
abline(h=0)
```



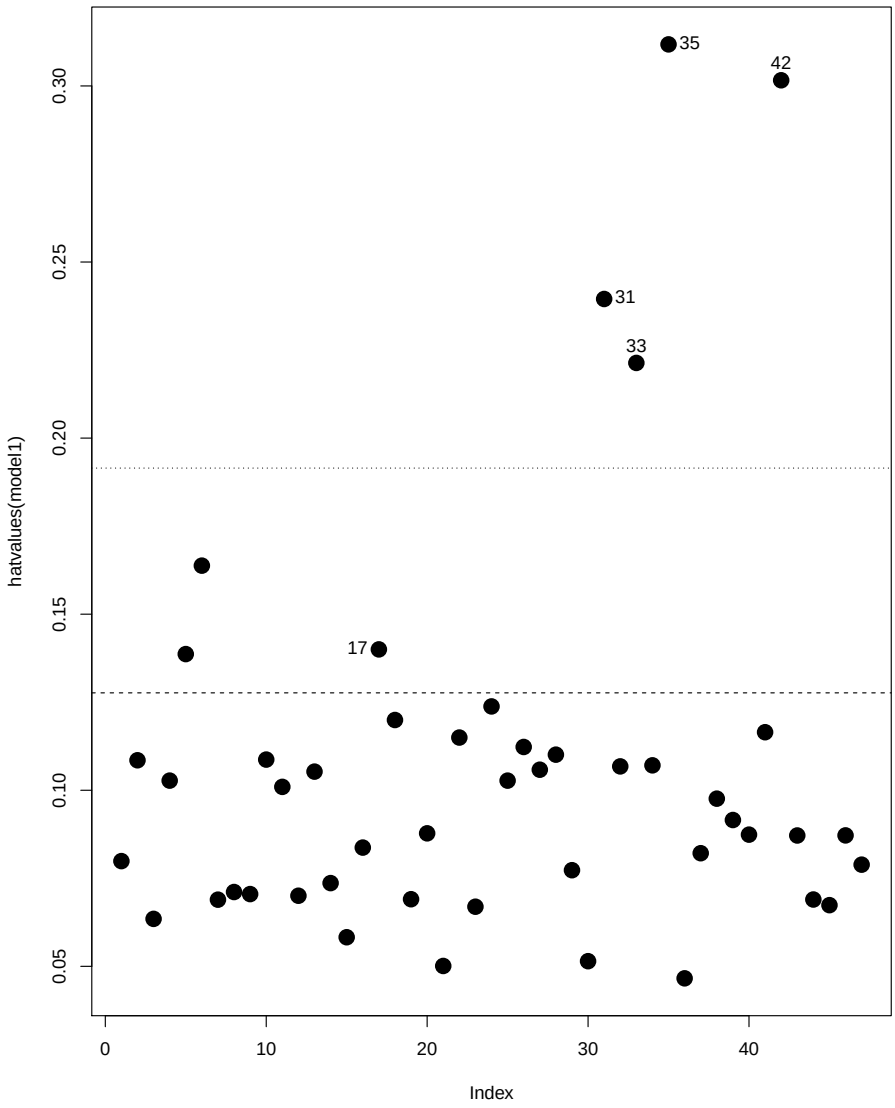
(b) Check the normality assumption with a Q-Q plot of the studentized residuals.

```r
#b)
qqPlot(model1)
```



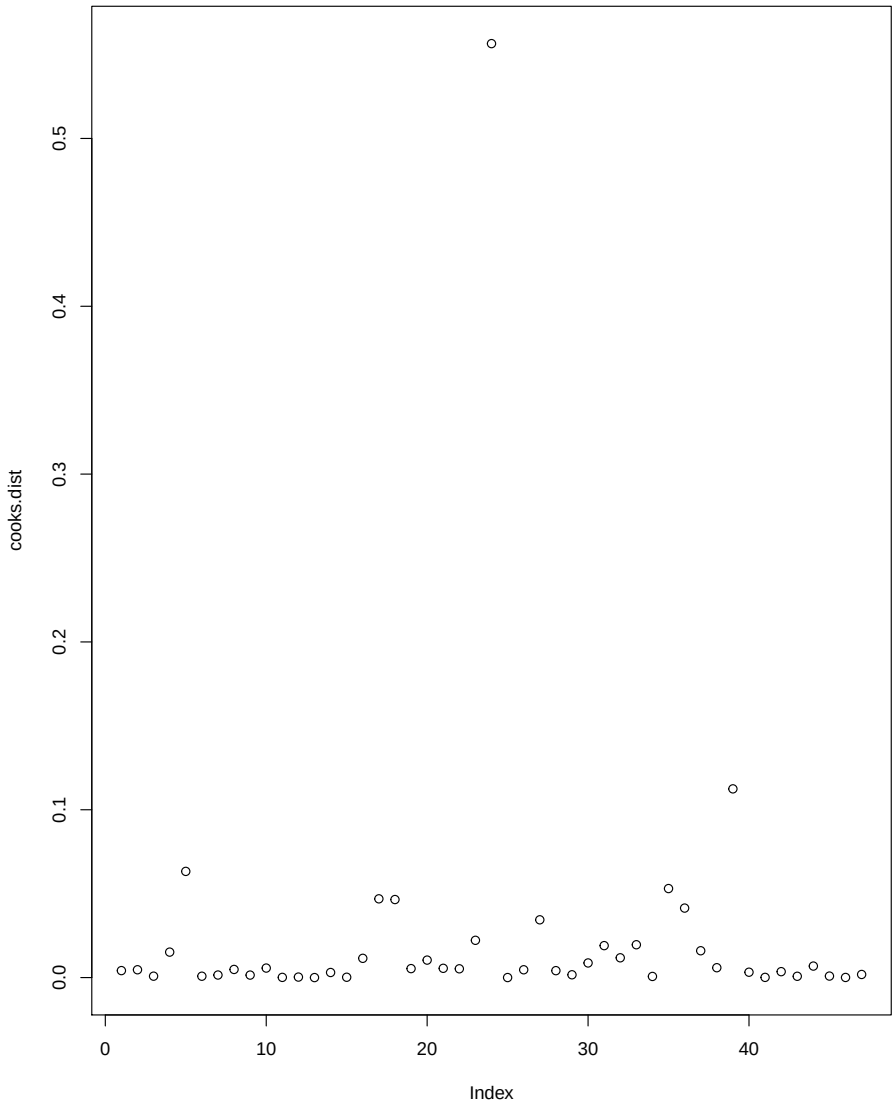
(c) Check for large leverage points by plotting the h values.

```r
#c)
hatvalues(model1)
plot(hatvalues(model1),pch=16,cex=2)
abline(h=2*3/47,lty=2)
abline(h=3*3/47,lty=3)
identify(1:47,hatvalues(model1),row.names(gamble))
```



(d) Check for outliers by running an `outlierTest`.

```r
#d)
outlierTest(model1)
cooks.dist <- cooks.distance(model1)
plot(cooks.dist)
```



Outlier Test Result:

rstudent : 24

unadjusted p-value: 6.016116

Bonferonni: 4.1041e-07

P: 1.9289e-05

(e) Check for influential points by creating a "Bubble plot" with the hat-values and studentized residuals.

```
#e)
influencePlot(model1,
              sub="Circle size is proportial to Cook's Distance")
```