

STAT 5310 Take-Home Test #2

Tom Wilson

November 7, 2018

1. (5 points)

Enter the variables X1, X2, and X3 into a multiple regression model predicting Y. Display the regression output.

```
data1 <- data.frame(Y = c(5,6,8,9,11)
                    ,X1 = c(1,200,-50,909,506)
                    ,X2 = c(1004,806,1058,100,505)
                    ,X3 = c(6,7.3,11,13,13.1)
                    )

fit <- lm(formula = Y ~ X1+X2+X3,data = data1)
summary(fit)

##
## Call:
## lm(formula = Y ~ X1 + X2 + X3, data = data1)
##
## Residuals:
##          1          2          3          4          5
## -2.493e-13  2.924e-13  3.733e-14 -3.893e-14 -4.144e-14
##
## Coefficients:
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept) -1.000e+03  2.728e-10 -3.665e+12 1.74e-13 ***
## X1           1.000e+00  2.728e-13  3.666e+12 1.74e-13 ***
## X2           1.000e+00  2.727e-13  3.667e+12 1.74e-13 ***
## X3           1.330e-14  2.156e-13  6.200e-02   0.961
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.902e-13 on 1 degrees of freedom
## Multiple R-squared:  1, Adjusted R-squared:  1
## F-statistic: 4.992e+25 on 3 and 1 DF, p-value: 1.04e-13
```

2. (15 points)

Create a table of R2adjusted, AIC, AICC, and BIC for the best subset of each size. Identify the optimal model or models from the approach based on all possible subsets.

```
Xs <- c('X1','X2','X3')
models <- interaction(Xs,Xs,Xs, sep='+') %>% levels() %>% paste("Y ~",.)

AICc_from_AIC <- function(AIC,fit) {
  n <- length(fit$residuals)
  k <- length(fit$coefficients) - 1
```

```

correction <- (2*k^2 + 2*k) / (n - k - 1)
return ( AIC + correction )
}

result <- data_frame()
for (form in models) {
  #print(form)
  fit <- lm(data=data1, formula = as.formula(form))
  glance_of_fit <- glance(fit) %>%
    mutate( model=form
            ,k = length(fit$coefficients) - 1
            ,AICc = AICc_from_AIC(AIC,fit)) %>%
    dplyr::select(c( 'model'
                    , 'k'
                    , 'adj.r.squared'
                    , 'AIC'
                    , 'AICc'
                    , 'BIC' )
                 )
  result <- rbind(result,glance_of_fit)
}

```

best metrics for each size

```

result %>% group_by(k) %>% summarise( max(adj.r.squared)
                                     ,min(AIC)
                                     ,min(AICc)
                                     ,min(BIC)
                                   ) %>% kable()

```

k	max(adj.r.squared)	min(AIC)	min(AICc)	min(BIC)
1	0.8764847	15.88064	17.21397	14.70895
2	1.0000000	-273.89202	-267.89202	-275.45426
3	1.0000000	-279.69529	-255.69529	-281.64810

best models for each metric

```

rbind(result[which.max(result$adj.r.squared),]
      ,result[which.min(result$AIC),]
      ,result[which.min(result$AICc),]
      ,result[which.min(result$BIC),]
    ) %>% kable()

```

model	k	adj.r.squared	AIC	AICc	BIC
Y ~ X2+X1+X1	2	1	-273.8920	-267.8920	-275.4543
Y ~ X3+X1+X2	3	1	-279.6953	-255.6953	-281.6481
Y ~ X2+X1+X1	2	1	-273.8920	-267.8920	-275.4543
Y ~ X3+X1+X2	3	1	-279.6953	-255.6953	-281.6481

3. (10 points)

Use the Forward Elimination method to determine the regression equation when starting with the same predictor variables listed in 1.

```
minimal.model <- lm(Y ~ 1, data = data1)
forward.step.model <- stepAIC( minimal.model
                              ,direction = "forward"
                              ,scope = list( upper = ~X1+X2+X3
                                             ,lower = ~ 1)
                              ,trace = TRUE)
```

```
## Start:  AIC=9.59
## Y ~ 1
##
##           Df Sum of Sq    RSS    AIC
## + X3       1   20.6879   2.1121 -0.3087
## + X1       1    8.6112  14.1888  9.2151
## + X2       1    8.5064  14.2936  9.2519
## <none>                22.8000  9.5866
##
## Step:  AIC=-0.31
## Y ~ X3
##
##           Df Sum of Sq    RSS    AIC
## <none>                2.1121 -0.30875
## + X2       1  0.066328  2.0458  1.53172
## + X1       1  0.064522  2.0476  1.53613
```

```
summary(forward.step.model )
```

```
##
## Call:
## lm(formula = Y ~ X3, data = data1)
##
## Residuals:
##      1      2      3      4      5
## 0.03434 0.13124 -0.43912 -0.82850  1.10203
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.7975      1.3452   0.593  0.5950
## X3            0.6947      0.1282   5.421  0.0123 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8391 on 3 degrees of freedom
## Multiple R-squared:  0.9074, Adjusted R-squared:  0.8765
## F-statistic: 29.38 on 1 and 3 DF,  p-value: 0.01232
```

4. (10 points)

Repeat Part 3, except use the Backward Elimination method.

Is the solution different from the one you got using the Forward method?

```

maximal.model <- lm(Y ~ X1+X2+X3, data = data1)
backward.step.model <- stepAIC( maximal.model
                                ,direction = "backward"
                                ,scope = list(upper = ~X1+X2+X3
                                              ,lower = ~ 1)
                                ,trace = TRUE)

## Start:  AIC=-285.77
## Y ~ X1 + X2 + X3

## Warning: attempting model selection on an essentially perfect fit is
## nonsense

##           Df Sum of Sq    RSS      AIC
## - X3       1     0.0000 0.0000 -287.749
## <none>                0.0000 -285.768
## - X1       1     2.0458 2.0458   1.532
## - X2       1     2.0476 2.0476   1.536
##
## Step:  AIC=-287.75
## Y ~ X1 + X2

## Warning: attempting model selection on an essentially perfect fit is
## nonsense

##           Df Sum of Sq    RSS      AIC
## <none>                0.000 -287.749
## - X2       1     14.189 14.189   9.215
## - X1       1     14.294 14.294   9.252

summary(backward.step.model )

##
## Call:
## lm(formula = Y ~ X1 + X2, data = data1)
##
## Residuals:
##          1          2          3          4          5
## -2.545e-13  2.854e-13  5.287e-14 -2.928e-14 -5.448e-14
##
## Coefficients:
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept) -1.000e+03  7.388e-11 -1.354e+13  <2e-16 ***
## X1           1.000e+00  7.312e-14  1.368e+13  <2e-16 ***
## X2           1.000e+00  7.339e-14  1.363e+13  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.764e-13 on 2 degrees of freedom
## Multiple R-squared:  1, Adjusted R-squared:  1
## F-statistic: 1.492e+26 on 2 and 2 DF, p-value: < 2.2e-16

```

5. (10 points)

Are different models chosen?

If so, carefully explain why different models are chosen.

Different models are found by forward and backward selection.

Forward selection finds $Y \sim X_3$ is the best $k=1$ model (minimizes AIC) subsequently, adding X_2 or X_1 does not decrease AIC.

backward selection finds that removing X_3 from the full model, $Y \sim X_1 + X_2 + X_3$ minimizes AIC and subsequently removing X_1 or X_2 does not improve AIC.

Decide on which model you would recommend.

At this point, describe and examine the assumptions of multiple linear regression for your recommended model.

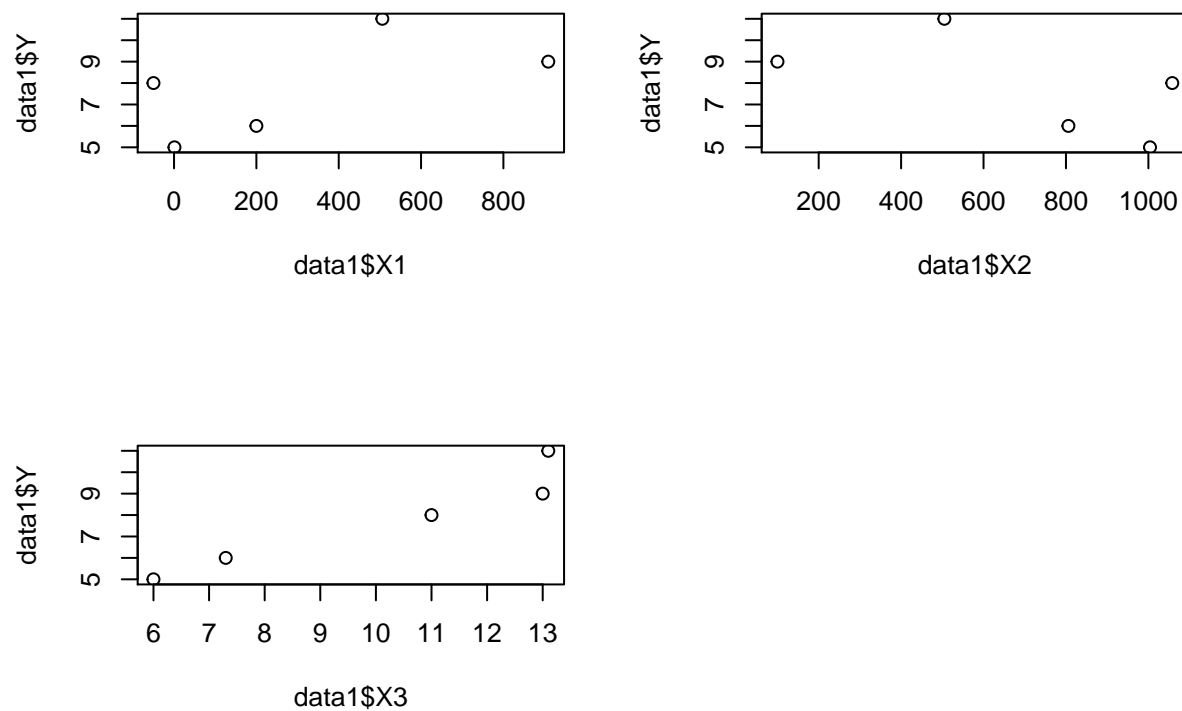
If any assumptions are violated – discuss what steps would/should be performed.

Checking Assumptions

Linear relationship

There is an assumption that Y is related to each x by the simple linear regression. We can confirm this with scatterplots.

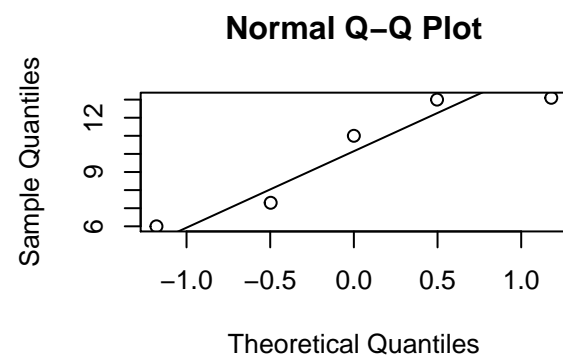
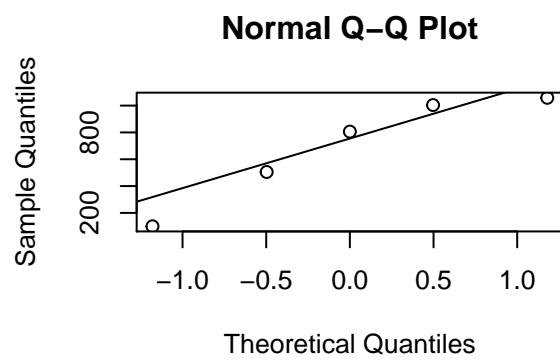
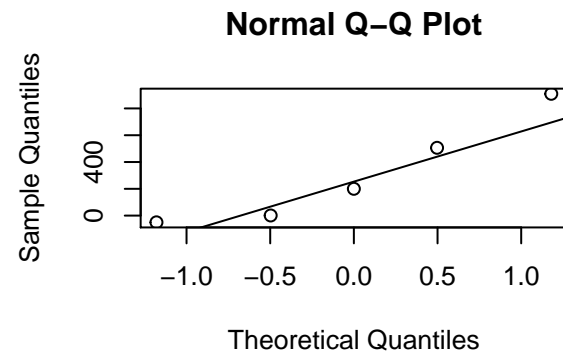
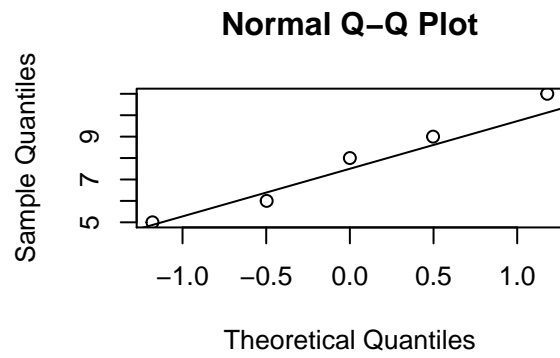
```
par(mfrow=c(2,2))
plot(data1$X1,data1$Y)
plot(data1$X2,data1$Y)
plot(data1$X3,data1$Y)
```



Visually, there is a plausible linear relationship between Y and each X.

Multivariate normality

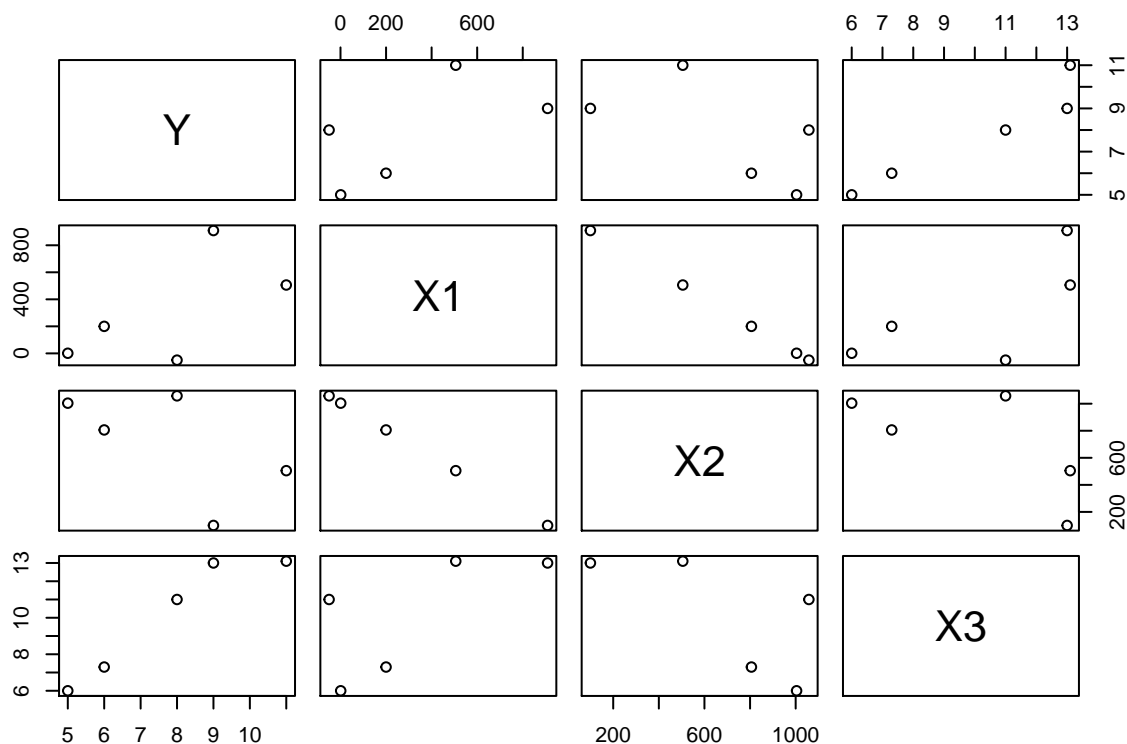
```
par(mfrow=c(2,2))
qqnorm(data1$Y)
qqline(data1$Y)
qqnorm(data1$X1)
qqline(data1$X1)
qqnorm(data1$X2)
qqline(data1$X2)
qqnorm(data1$X3)
qqline(data1$X3)
```



visually, each X and Y appear to be normally distributed.

No or little multicollinearity

```
pairs(data1[c('Y', 'X1', 'X2', 'X3')])
```



```
cor(data1)
```

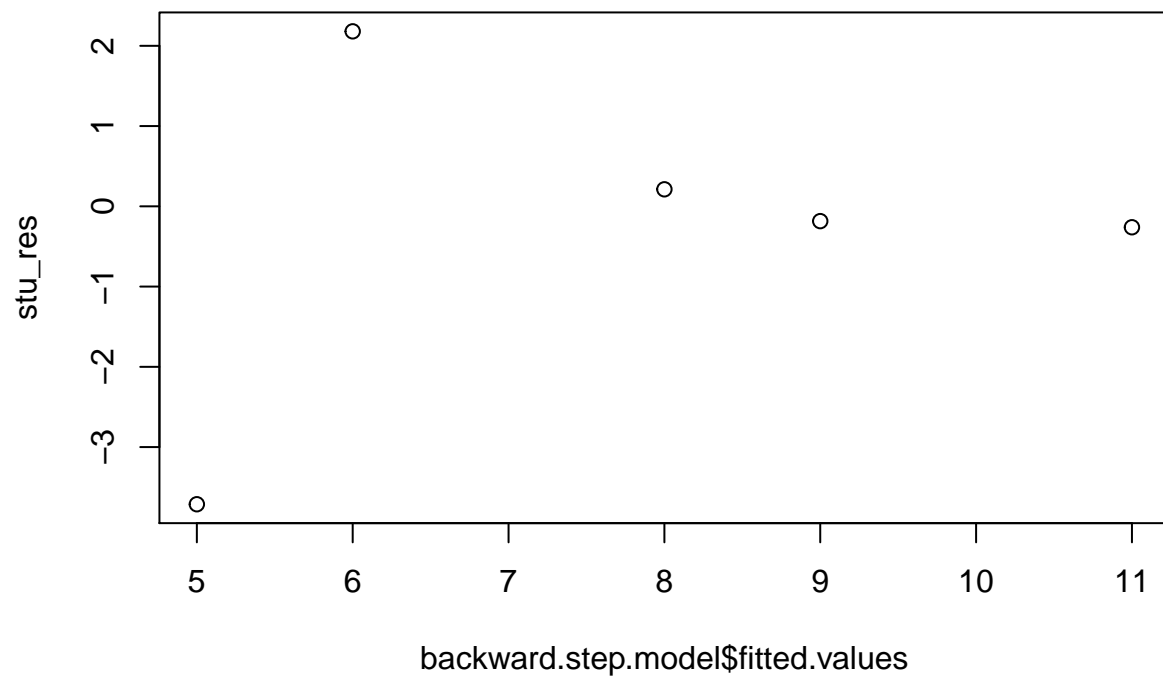
```
##           Y           X1           X2           X3
## Y    1.0000000  0.6145611 -0.6108095  0.9525563
## X1  0.6145611  1.0000000 -0.9999887  0.6858141
## X2 -0.6108095 -0.9999887  1.0000000 -0.6826107
## X3  0.9525563  0.6858141 -0.6826107  1.0000000
```

X1 and X2 are highly correlated. This violates the assumption of no multicollinearity. We should not consider models which include both X1 and X2.

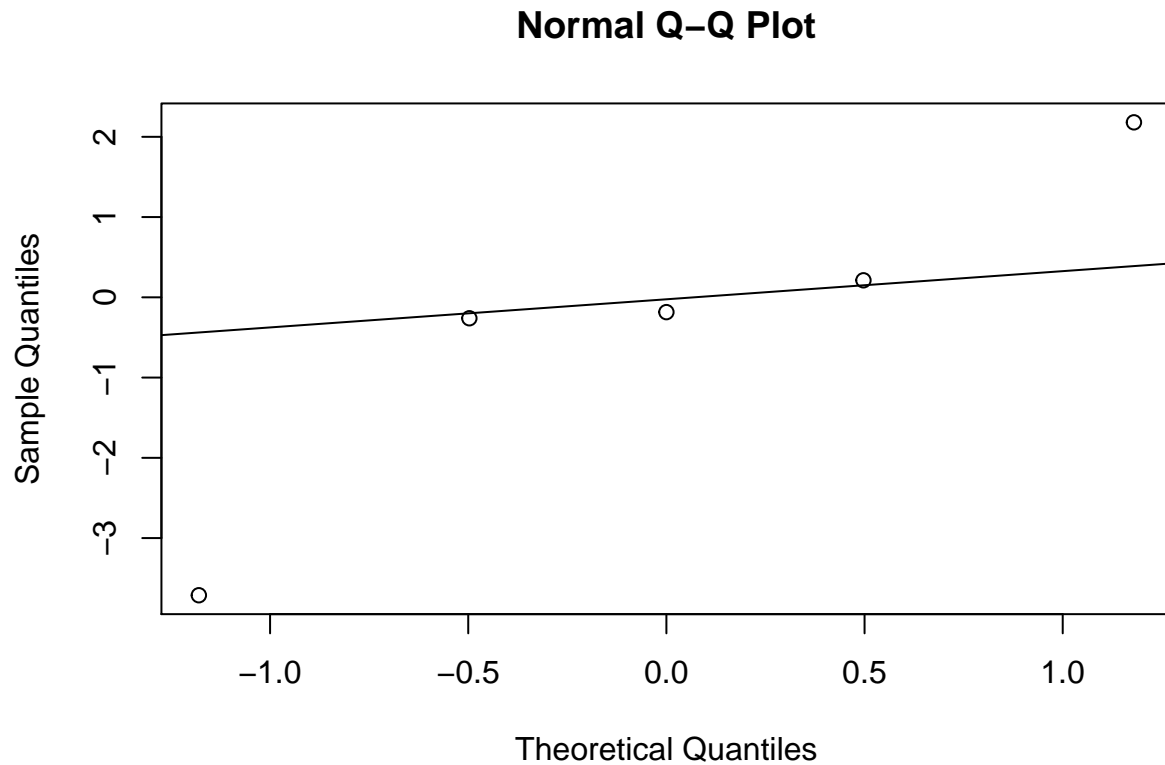
Homoscedasticity

This assumption means that the variance is equal. Moreover, the residuals are independent of each other, and are identically normally distributed with a mean of 0 and variance of σ^2 . We can confirm this by inspecting the residuals.

```
stu_res <- studres(backward.step.model)
plot(backward.step.model$fitted.values, stu_res)
```

```
qqnorm(stu_res)  
qqline(stu_res)
```



6. (50 points)

Using the sheet/page “2010 to 2013 Wide Release Movies” from the CreditCard dataset of Lab/Homework #3.

Recommend a model to predict the y variable – “Opening Weekend Gross” with the possible predictors (no interactions) – Runtime, Production Budget, Critic Rating, Audience Rating, and/or Month of Release.

Explain how you determined your model and why you recommended it over other models.