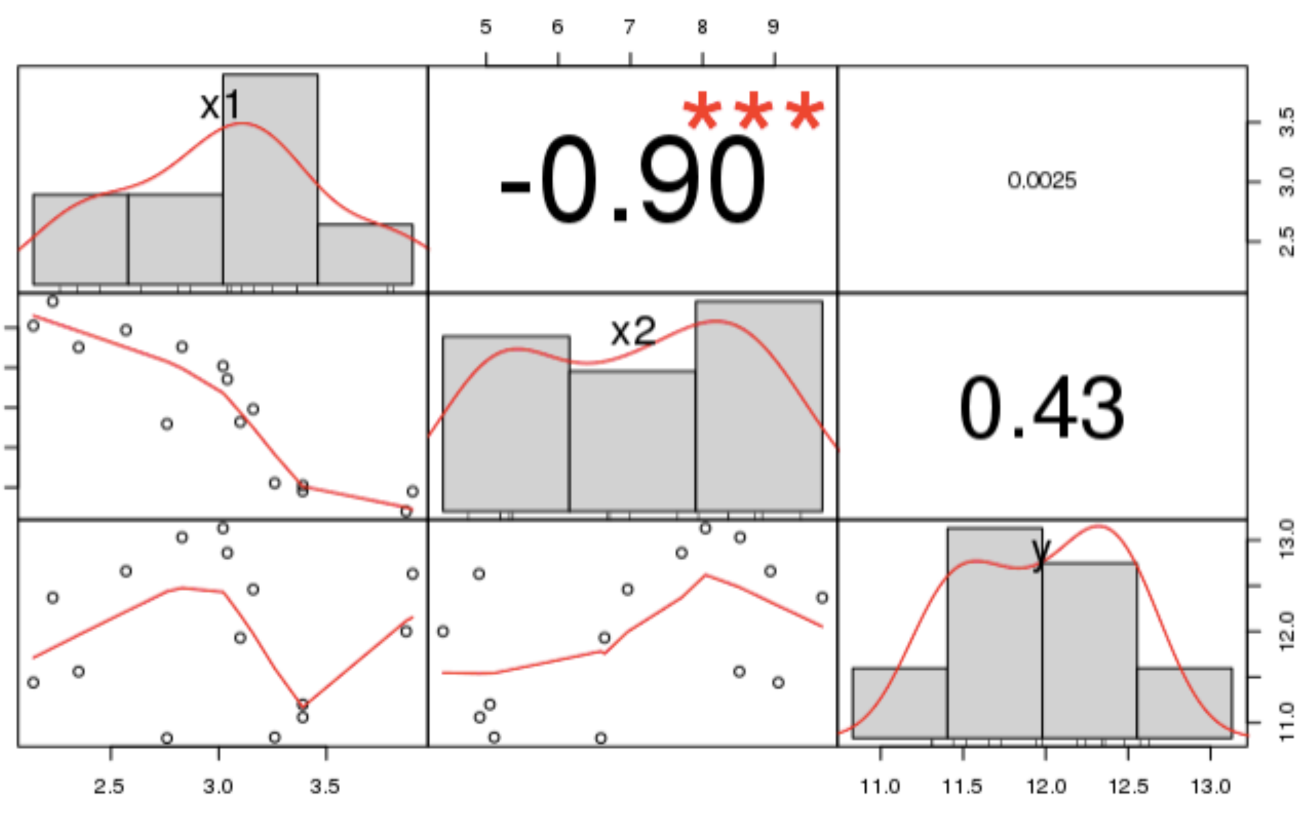
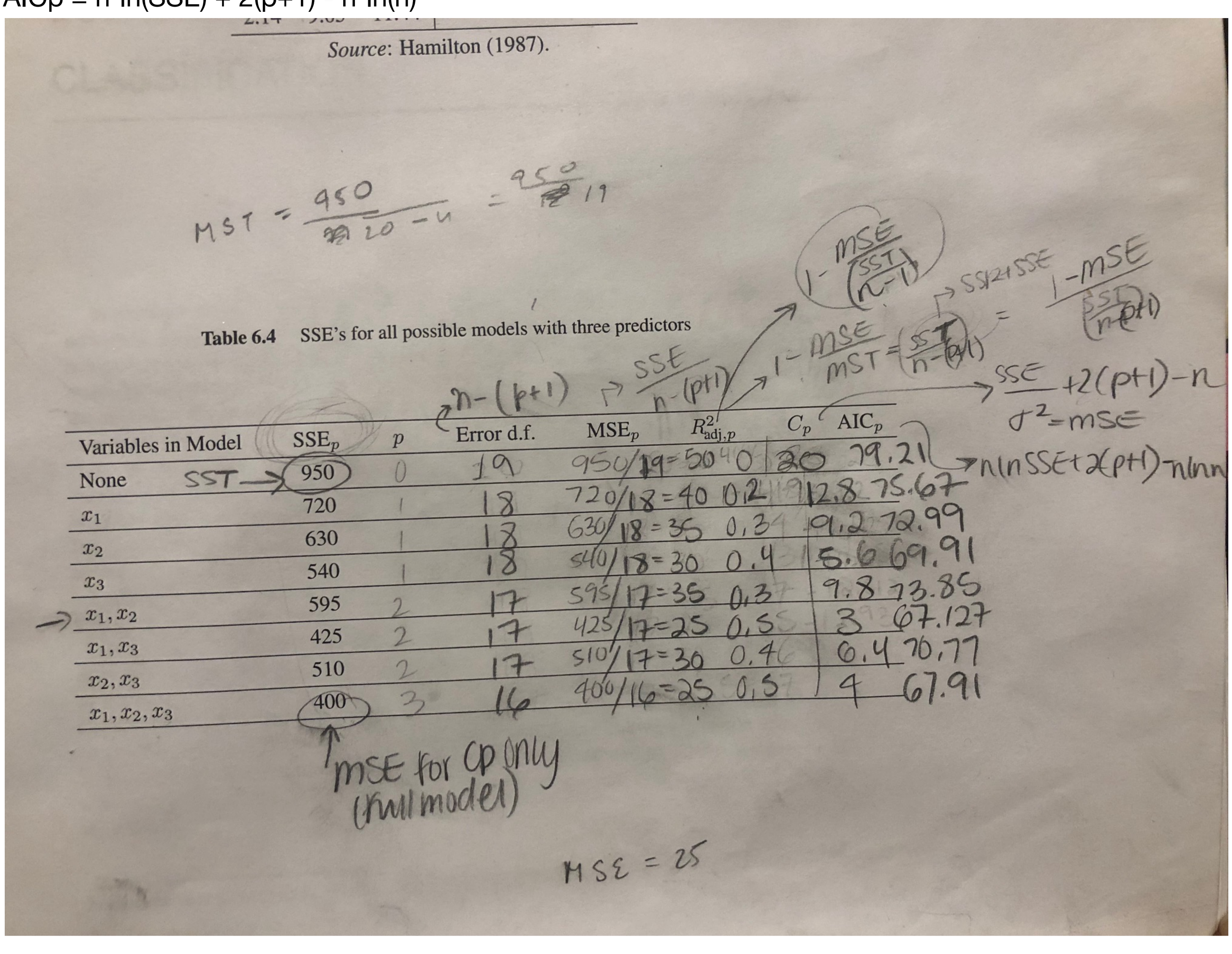
Naomi Kaduwela
Nov 9 2018

6.2

```r
#6.2

# Import and store data
hamilton <- read.csv("~/Predictive Analytics/Hamilton.csv", stringsAsFactors = FALSE)
View(hamilton)
# a.) Matrix Scatter Plot & Pairwise Correlation
# x1 and x2 seem to have a strong negative correlation (-.9)
  # X's being related could be an indicator of multicollinearity
# x2 is somewhat positively correlated with y (0.43)
# x1 is not very correlated with y (0.0025)

#pairs(hamilton[,1:3], pch = 19)
chart.Correlation(hamilton, histogram=TRUE, pch=19)

# b.) Regress y on x1x2. What is R^2? How do you explain R^2 being close to 1 when both x1 and x2 have low correlations with y?
# R^2 = 0.9998. As we can see, alone x1 and x2 have low correlations with y. Together, they are good predictors of y because
# together they are linear combination is strongly correlated with y
hamiltonLM <- lm(hamilton$y ~ hamilton$x1 + hamilton$x2)
summary(hamiltonLM)

# c.) Why is it that: Forward step wise = wrong but Backward step wise = correct
# Because the forward step wise starts with the NULL model and neither x1 or x2 are individually highly correlated with y,
# we might end up with a NULL model
# If we start with the backwards model, we will start with both x1 and x2 in the model, which will result in the high r^2 result
```



6.3
a.)
p = number of predictors
df = n- (p+1)
MSE = SSE/n-(p+1)
R^2adj = 1- MSE/MST = 1-MSE / (SST/n-1)
  *Where SST/n-1 = 950/19 for all
Cp = SSE/SD^2 + 2(p+1) - n = SSE/MSE + 2(p+1) - n
  *Where MSE = 400/16 = 25 for all denominators
AICp = n*ln(SSE) + 2(p+1) - n*ln(n)



Source: Hamilton (1987).

MST = 950/(20 - 1) = 950/19

Table 6.4 SSE's for all possible models with three predictors

Variables in Model	SSE_p	p	Error d.f.	MSE_p	$R^2_{adj,p}$	C_p	AIC_p
None	950	0	19	950/19 = 50	0	20	79.21
x_1	720	1	18	720/18 = 40	0.2	12.8	75.67
x_2	630	1	18	630/18 = 35	0.3	9.2	72.99
x_3	540	1	18	540/18 = 30	0.4	5.6	69.91
x_1, x_2	595	2	17	595/17 = 35	0.3	9.8	73.85
x_1, x_3	425	2	17	425/17 = 25	0.5	3	67.127
x_2, x_3	510	2	17	510/17 = 30	0.4	6.4	70.77
x_1, x_2, x_3	400	3	16	400/16 = 25	0.5	4	67.91

MSE for Cp only (full model)

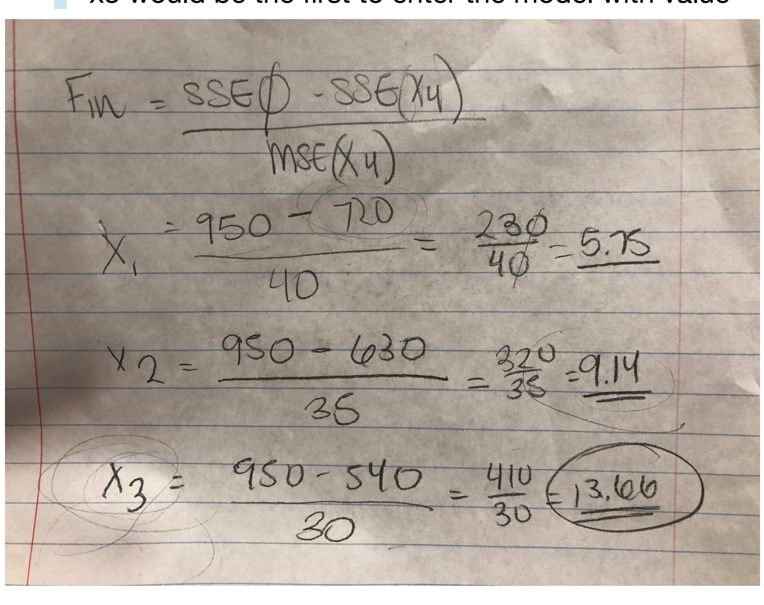
MSE = 25

b.)
All models choose x1,x3

- R^2 adj maximizes so the best model would be 0.5. Since we have 2 models with 0.5 x1,x3 and x1,x2,x3 we would select the model with less parameters: x1,x3
- Cp minimizes so the best model would be 3 with x1,x3
- AICp minimizes so the best model would be 67.127 with x1,x3

c.) Stepwise regression: Fin = Fout = 4, which variable would be the first to enter and what is the Fin value?

- x3 would be the first to enter the model with value = 13.66



$$F_{in} = \frac{SSE(\emptyset) - SSE(X_4)}{MSE(X_4)}$$

$$X_1 = \frac{950 - 720}{40} = \frac{230}{40} = 5.75$$

$$X_2 = \frac{950 - 630}{35} = \frac{320}{35} = 9.14$$

$$X_3 = \frac{950 - 540}{30} = \frac{410}{30} = 13.66$$

d.) The second variable to enter the equation will be x1 because the Fin value = 4.6. This is > 4 (the Fin criteria) and has the highest partial correlation coefficient.

- (540 -425) / 25 = 4.6

Partial correlation coeff = 0.46

- sqrt(SSE(x3) - SSE(x1,x3)/SSE(x3))
- sqrt(540-425/540) = 0.46

e.) Fout = SSE(x1) -SSE(x1x3)/MSE(x1x3) = 720 - 425 / 25 = 11.8
  Since 11.8 > 4 we keep the second variable x1

f.) SSE(x1,x3) - SSE (x1,x2,x3) / MSE(x1,x2,x3) = 1
  Because 1 < 4 we will keep the model as is and not add the x2 to go to the full model

6.4
Here we see the SSE values are close, with model 1 being slightly higher, making it better.
model 1: 0.6339853
model 2: 0.6430125
However, even though model 2 has a higher SSE, we might still choose it because it has more significant predictors

Model 1

```r
#6.4
# Import and store data
car <- read.csv("~/Predictive Analytics/carprices.csv", stringsAsFactors = FALSE)
car

#Combine variables that are not included in the model
car$Make[car$Make == "Pontiac"] <- "Combined"
car$Make[car$Make == "Saturn"] <- "Combined"
#Fix mileage since it's in thousandths
car$Mileage  <- car$Mileage/ 1000

# relevel data
car$Make <- relevel(as.factor(car$Make), ref = "Combined")
car$Type <- relevel(as.factor(car$Type), ref = "Wagon")

#split data into test and training set
oddTraining <- car[ c(TRUE,FALSE), ] #odd
evenTest <- car[ c(FALSE,TRUE), ] #even

#Model 1 from 6.3 Example
oddTrainingLM <- lm(log10(Price) ~ Mileage + Cylinder + Liter + Cruise + Type + Make, data = oddTraining)
summary(oddTrainingLM)

#Calculate SSE -- 0.634
output <- predict(oddTrainingLM, evenTest)
res1 <- (output - log10(evenTest$Price))^2
sse1 <- sum(res1)
sse1
```

```
-0.138532 -0.024858  0.002488  0.025478  0.114801

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.0778193  0.0163209 249.853  < 2e-16 ***
Mileage         -0.0034849  0.0002387 -14.597  < 2e-16 ***
Cylinder        -0.0099119  0.0066279  -1.495   0.1356
Liter            0.1068840  0.0075944  14.074  < 2e-16 ***
Cruise           0.0069594  0.0055922   1.244   0.2141
TypeConvertible  0.0712222  0.0115543   6.164 1.77e-09 ***
TypeCoupe       -0.0665500  0.0095425  -6.974 1.33e-11 ***
TypeHatchback   -0.0847389  0.0116383  -7.281 1.85e-12 ***
TypeSedan       -0.0697423  0.0081568  -8.550 2.85e-16 ***
MakeBuick        0.0349741  0.0076989   4.543 7.42e-06 ***
MakeCadillac     0.2322969  0.0093297  24.899  < 2e-16 ***
MakeChevrolet   -0.0150798  0.0053899  -2.798   0.0054 **
MakeSAAB         0.2789042  0.0080552  34.624  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03919 on 389 degrees of freedom
Multiple R-squared:  0.9528,    Adjusted R-squared:  0.9514
F-statistic: 654.6 on 12 and 389 DF,  p-value: < 2.2e-16

[1] 0.6339853
```

Model 2

```r
#Model 2 from 6.6 Example

car <- read.csv("~/Predictive Analytics/carprices.csv", stringsAsFactors = FALSE)
car

#Combine variables that are not included in the model
car$Type[car$Type == "Sedan"] <- "Combined"
car$Type[car$Type == "Coupe"] <- "Combined"
car$Type[car$Type == "Hatchback"] <- "Combined"
#Fix mileage since it's in thousandths
car$Mileage  <- car$Mileage/ 1000

# relevel data
car$Make <- relevel(as.factor(car$Make), ref = "Buick")
car$Type <- relevel(as.factor(car$Type), ref = "Combined")

#split data into test and training set
oddTraining <- car[ c(TRUE,FALSE), ] #odd
evenTest <- car[ c(FALSE,TRUE), ] #even

oddTrainingLM <- lm(log10(Price) ~ Mileage + Cylinder + Liter + Type + Make, data = oddTraining)
summary(oddTrainingLM)

#Calculate SSE -- 0.6430125
output <- predict(oddTrainingLM, evenTest)
res1 <- (output - log10(evenTest$Price))^2
sse1 <- sum(res1)
sse1
```

```
Residuals:
      Min        1Q    Median        3Q       Max
-0.138109 -0.022875  0.001581  0.024468  0.118753

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.0608055  0.0176649 229.880  < 2e-16 ***
Mileage         -0.0034806  0.0002393 -14.545  < 2e-16 ***
Cylinder        -0.0167644  0.0062650  -2.676  0.00777 **
Liter            0.1151789  0.0069327  16.614  < 2e-16 ***
TypeConvertible  0.1410746  0.0093542  15.081  < 2e-16 ***
TypeWagon        0.0651714  0.0083525   7.803 5.59e-14 ***
MakeCadillac     0.2013134  0.0100593  20.013  < 2e-16 ***
MakeChevrolet   -0.0552295  0.0072815  -7.585 2.45e-13 ***
MakePontiac     -0.0322131  0.0078731  -4.092 5.21e-05 ***
MakeSAAB         0.2439345  0.0096750  25.213  < 2e-16 ***
MakeSaturn      -0.0449952  0.0101659  -4.426 1.25e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0393 on 391 degrees of freedom
Multiple R-squared:  0.9523,    Adjusted R-squared:  0.9511
F-statistic: 780.8 on 10 and 391 DF,  p-value: < 2.2e-16

[1] 0.6430125
```