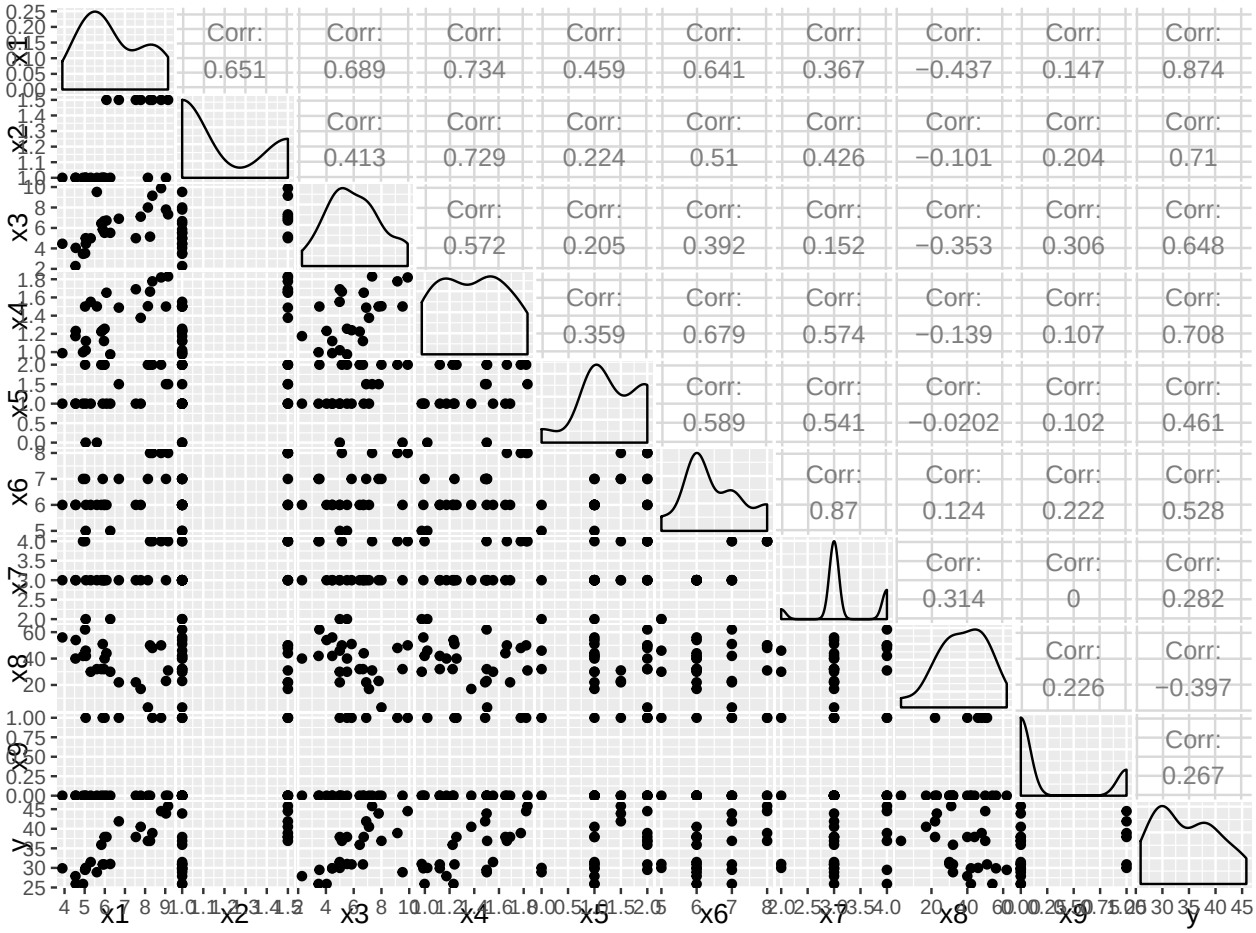
Property

Cody Frisby

March 31, 2016

1) Scatterplot and Variance Inflation Factors

Here, I produce a scatterplot matrix of all the variables.



Variance inflation factors are

x1	7.021892
x2	2.835480
x3	2.454864
x4	3.836332
x5	1.823414
x6	11.710866
x7	9.721847
x8	2.321052
x9	1.942424

It appears **x6** and **x7** have colleniarity to be concerned of. We should exclude **x6** from the model. When we do our variance inflation factors are

x1	4.723504
x2	2.593827
x3	2.437637
x4	3.827800
x5	1.819725
x7	2.848920
x8	2.320091
x9	1.496295

and this looks much better now.

2) Best Model Forward Selection

Stepping forward, using R, and AICs, the best model includes **x1** and **x2**.

3) Best Model Backward Selection

Stepping Backward, using R, and AICs, the best model also includes **x1** and **x2**. But, if we do not exclude **x7** (vif = 9.72185) then we arrive at a different model, one with 4 covariates, namely **x1**, **x2**, **x5**, and **x7**.

4) Best Model Stepwise Selection

Stepwise, using R, stepAIC() function from MASS package, starting with the full model (excluding x6) gives us the same model as no. 3 above, namely **x1**, **x2**, **x5**, and **x7**.

5) All Subsets, Top 30

rankBIC	x1	x2	x3	x4	x5	x7	x8	x9
1	*	*						
2	*							
3	*	*						*
4	*							*
5	*	*			*	*		
6	*	*				*		
7	*	*			*			
8	*	*	*					
9	*	*					*	
10	*			*				
11	*	*		*				
12	*	*					*	*
13	*				*			
14	*		*					
15	*					*		
16	*						*	
17	*			*				*
18	*	*				*		*

rankBIC	x1	x2	x3	x4	x5	x7	x8	x9
19	*	*			*			*
20	*	*			*	*		*
21	*	*	*		*	*		
22	*	*			*		*	
23	*	*		*	*	*		
24	*	*	*		*			
25	*						*	*
26	*				*			*
27	*	*			*		*	*
28	*	*	*					*
29	*	*		*				*
30	*					*		*

For this question, using R, we use the leaps() funciton from the leaps package. I wrote some R code to do this for me. The predictors I'd choose, based on the lowest AIC, BIC, Cp, and parsimony, are **x1** and **x2**. Additionaly, I display the matrix above, which has been sorted by lowest BIC, asteriks indicating which variables are in the model. Interestingly, when we stepped backward and used stepwise the model we chose using those two methods is the 5th best here.

6) Chosen Model

My chosen model is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$$

where β_0 is the expected sale price (in thousands) of a home when there are zero taxes and zero bathrooms, β_1 is the expected change in sale price per unit increase in property taxes (x_1, in thousands), β_2 is the expected increase in sale price for each additional bathroom (x_2). Here, $1 \leq x_2 \leq 1.5$.

7) Fit the model

Fit the model from above here

$$SalePrice = 10.1120314 + 2.7170257 x_1 + 6.0985076 x_2$$

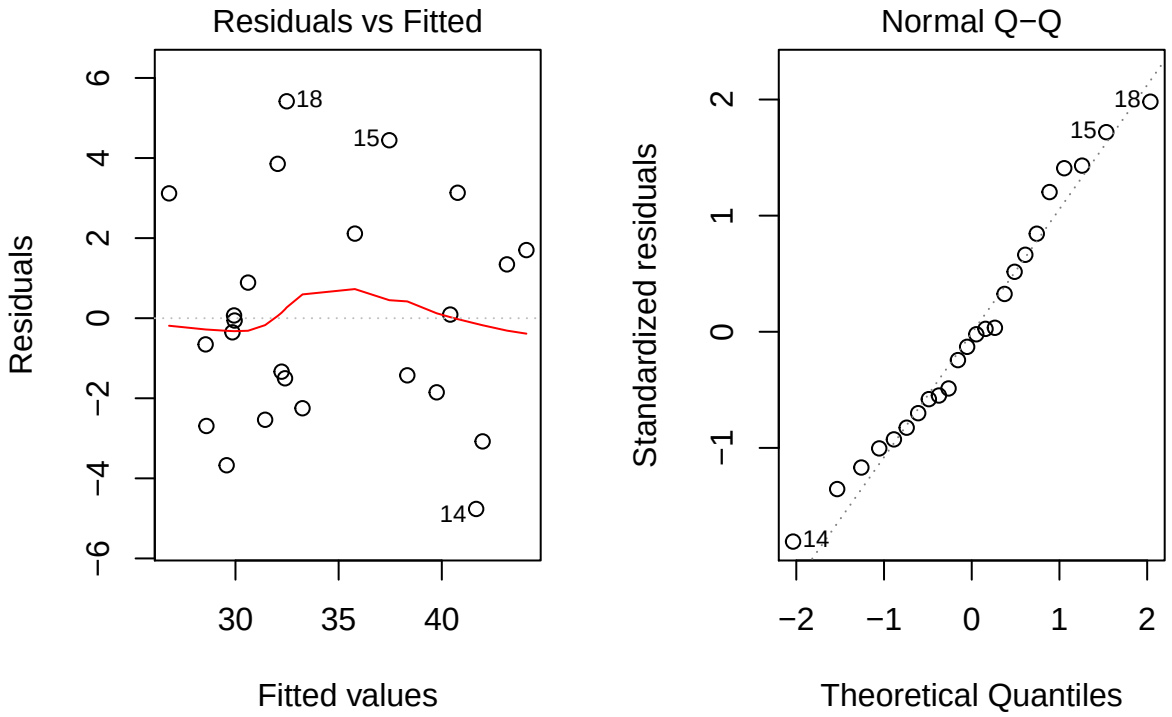
and also display 95% confidence intervals

	2.5 %	97.5 %
(Intercept)	3.8812157	16.342847
x1	1.6956321	3.738419
x2	-0.6125206	12.809536

meaning for every $1000 inrease in a homes' property taxes we expect the sale price to increase by a factor of 1.696. Additionaly, the interval for x_2 contains 0, this shows that x_2 may not be significant, although I have chosen to include it since it provides additional explanation beyond only x_1.

8) Model Fit and Diagnostics

Look first at the model residual plots



These two plots look good and there is nothing to be concerned about regarding the model assumptions. The model R^2 is equal to 0.7980639 and is interpreted in this context as being able to explain approximately 79% of the variation in sales price by knowing the taxes and number of bathrooms of the home.

9) SLR Model

The confidence intervals for the simple linear regression model

	2.5 %	97.5 %
(Intercept)	7.972598	18.737995
x1	2.504646	4.138379

and some of the summary statistics of the model

sigma	R.sq	AIC	BIC
2.988369	0.7637216	124.5677	128.1019

10) Conclusions

The most important variables for predicting the sale price of a home appear to be the property taxes and the number of bathrooms the home has.
The claims of the real estate agent appear to be somewhat valid, although our prediction accuracy actually increases when we include the number of bathrooms of the home. This is evidence by a smaller value for sigma as well as lower values for AIC and BIC. Here I display σ from the two models.

sigma from chosen model	sigma from taxes only model
2.827685	2.988369

Although these two values for RMSE are very close, I'd go with the model that has the smaller value, since our predictions will be more accurate.

R code:

```
# get the data into R
prop <- xlsx::read.xlsx(
  "~/Documents/MATH3710/ProblemSets/problem7/property valuation.xlsx",
  sheetIndex = 1)
colnames(prop) <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "y")
predictors <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9")
fit.all <- lm(y ~ ., data = prop)
f <- formula(y ~ x1+x2+x3+x4+x5+x6+x7+x8+x9)
fit.null <- lm(y ~ 1, data = prop)
# top models 1:10 based on lowest BICs and AICs
fit1 <- lm(y ~ x1+x2, data = prop)
fit2 <- lm(y ~ x1, data = prop)
fit3 <- lm(y ~ x1+x2+x9, data = prop)
fit4 <- lm(y ~ x1+x9, data = prop)
fit5 <- lm(y ~ x1+x2+x5+x7, data = prop)
fit6 <- lm(y ~ x1+x2+x7, data = prop)
fit7 <- lm(y ~ x1+x2+x5, data = prop)
fit8 <- lm(y ~ x1+x2+x3, data = prop)
fit9 <- lm(y ~ x1+x2+x8, data = prop)
fit10 <- lm(y ~ x1+x4, data = prop)
mod.list <- list(fit1,fit2,fit3,fit4,fit5,fit6,fit7,fit8,fit9,fit10)
# fit the model without x6 and x7
step(fit.null, scope = list(upper=fit8), direction = "forward")
# the above code steps forward from ybar to find the best model.
# note, fit2 has the lowest AIC of any model
step(fit7, direction = "backward") #backward
# stepwise below
MASS::stepAIC(fit8, direction = "both")
# here we get the top 30 models based on Cp.
# we could use adj.r2 or r2 as well
mods.cp <- leaps::leaps(x = prop[,c(1:5,7:9)], y = prop[,10], method="Cp",
                        names = names(prop[,c(1:5,7:9)]), nbest=10)
ord <- order(mods.cp$Cp)
top30 <- ord[1:30]
mods30.cp <- mods.cp$which[top30, ]
# use adjusted R^2 now
mods.adjr2 <- leaps::leaps(x = prop[,c(1:5,7:9)], y = prop[,10],
                           method="adjr2",
                        names = names(prop[,c(1:5,7:9)]), nbest=10)
ord <- order(mods.adjr2$adjr2, decreasing = TRUE) # want large values first
top30 <- ord[1:30]
mods30.adjr2 <- mods.adjr2$which[top30, ]
# or we could do something easier below
```

```r
regs <- leaps::regsubsets(y ~ x1+x2+x3+x4+x5+x7+x8+x9, data=prop, nbest = 5)
regs.data <- as.matrix(cbind(summary(regs)$rsq, summary(regs)$adjr2,
                  summary(regs)$cp, summary(regs)$bic))
colnames(regs.data) <- c("r2", "adjr2", "cp", "bic")
# now we have a matrix with the above values
tmp <- summary(regs)$outmat # matrix of our models
par(mfrow=c(1,2))
plot(fit1, which = c(1,2))
r.sq <- summary(fit1)$r.squared; s1 <- summary(fit1)$sigma
s2 <- summary(fit2)$sigma; r.sq2 <- summary(fit2)$r.squared
aic2 <- AIC(fit2); bic2 <- BIC(fit2)
fit2.s <- cbind(s2,r.sq2, aic2, bic2)
colnames(fit2.s) <- c("sigma", "R squared", "AIC", "BIC")
x <- data.frame(x1 = 6, x2 = 1.25)
pred1 <- predict(fit1, newdata = x, se.fit = TRUE)
pred2 <- predict(fit2, newdata = x, se.fit = TRUE)
```