INSH5301 Intro Computational Statistics - Final Exam

Ali Banijamali

04/21/2020

Loading Libraries and Data :

```
library(Ecdat) # For housing data set
library(AER)   # For health insurance data set
library(ggplot2)
library(glmnet)
library(e1071)
library(stargazer)
library(knitr)
library(kableExtra)
library(car)
library(psych)
library(dplyr)

data(Housing)
kable( head(Housing, 2) , booktabs=T, caption='Housing Dataset') %>%
  kable_styling( latex_options=c('scale_down', 'hold_position', 'striped'))
```

Table 1: Housing Dataset

price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
42000	5850	3	1	2	yes	no	yes	no	no	1	no
38500	4000	2	1	1	yes	no	no	no	no	0	no

```
data(HealthInsurance)
kable( head(HealthInsurance, 2) , booktabs=T, caption='Health Insurance Dataset') %>%
  kable_styling( latex_options=c('scale_down', 'hold_position', 'striped'))
```

Table 2: Health Insurance Dataset

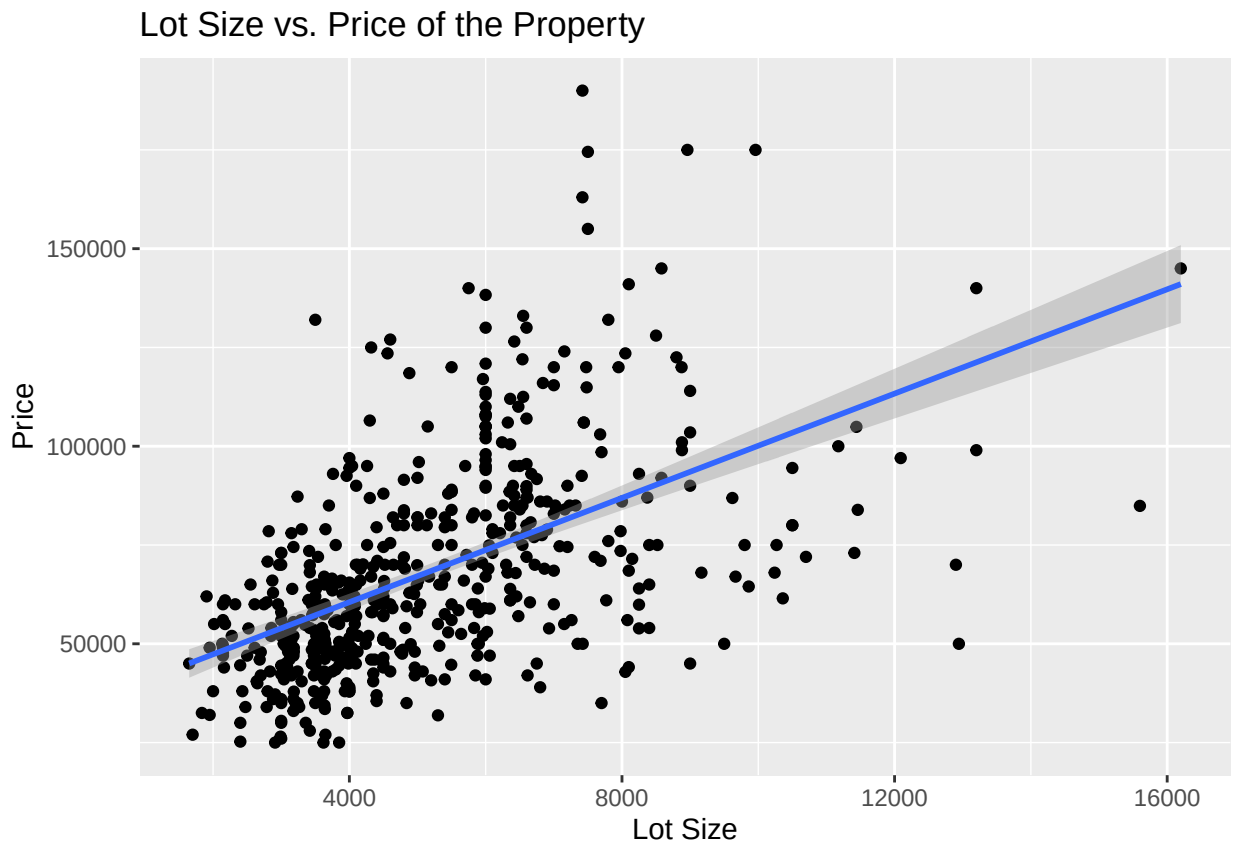
health	age	limit	gender	insurance	married	selfemp	family	region	ethnicity	education
yes	31	no	male	yes	yes	yes	4	south	cauc	bachelor
yes	31	no	female	yes	yes	no	4	south	cauc	highschool

1. *Bivariate Regression*

1. Using the Housing dataset, create a scatter plot of sale price of a house (y-axis) and the lot size of the property (x-axis). Use the ggplot function and include a regression line. Using the graph, describe the relation between the two variables.

ANS.

```
ggplot(data=Housing, aes(x=lotsize, y=price)) + geom_point() + geom_smooth(method=lm) +
  xlab('Lot Size') + ylab('Price') +
  ggtitle('Lot Size vs. Price of the Property')
```



According to the plot, there is a positive relationship between Lot Size and the price of the property. Meaning, as the size of the lot increases, the price of the property goes up.

2. Estimate a bivariate regression of the sale price of a house on the lot size of the property. Interpret the estimated β parameters, the statistical significance and R^2.

ANS.

```
bivar.model <- lm(price ~ lotsize, data=Housing)

stargazer(bivar.model, align=T, no.space=T, header=F,
          title="Price of the Property vs. Lot Size",
          dep.var.labels=c("Price"),
          covariate.labels=c("Lot Size"),
          omit.stat=c("LL", "ser", "f"),
          table.placement = "H" # TO hold the table at it's place (Not floating)
          # Floting tables get out of order in the printed output
         )
```

Table 3: Price of the Property vs. Lot Size

	Dependent variable:
	Price
Lot Size	6.599^{***}
	(0.446)
Constant	$34,136.190^{***}$
	(2,491.064)
Observations	546
R^2	0.287
Adjusted R^2	0.286
Note:	*p<0.1; **p<0.05; ***p<0.01

$\beta = +\ 6.599$ This shows a positive correlation, i.e. as the lotsize of the houses increase, their price goes up. The estimated β is statistically significant (p-value<2.2e-16) and the adjusted R^2 which almost equals to the R^2 in our case, because we only have one independent variable, is 0.286. meaning the independent variable (lot size), explains ~%29 of the variations in the house's price. β also tells us that each square foot increase in the lotsize, increases the price of the house by ~6.6:

$$Price = 6.599(lotsize) + 34,136.190$$

3. Is there any reason to believe that the estimated slope parameter in the previous regression is biased? (Explain)

ANS.
I think the lot size of the house is a reasonable variable for explainging the price of the property by itself, but let's look at other variables in our data:

```
colnames(Housing)
```

```
##  [1] "price"    "lotsize"  "bedrooms" "bathrms"  "stories"  "driveway"
##  [7] "recroom"  "fullbase" "gashw"    "airco"    "garagepl" "prefarea"
```

First, Let's think of exogeneity. When exogeneity is violated, our model will suffer from the Omitted Variable Bias (OVB). I will get a regression with all variables:

```
ex.check <- lm(price ~ ., data=Housing)
stargazer(ex.check, align=T, no.space=T, header=F,
          title="Price of the Property vs. all vars",
          dep.var.labels=c("Price"),
          omit.stat=c("LL", "ser", "f"),
          table.placement = "H")
```

Table 4: Price of the Property vs. all vars

	Dependent variable:
	Price
lotsize	3.546^{***}
	(0.350)
bedrooms	$1,832.003^{*}$
	(1,047.000)
bathrms	$14,335.560^{***}$
	(1,489.921)
stories	$6,556.946^{***}$
	(925.290)
drivewayyes	$6,687.779^{***}$
	(2,045.246)
recroomyes	$4,511.284^{**}$
	(1,899.958)
fullbaseyes	$5,452.386^{***}$
	(1,588.024)
gashwyes	$12,831.410^{***}$
	(3,217.597)
aircoyes	$12,632.890^{***}$
	(1,555.021)
garagepl	$4,244.829^{***}$
	(840.544)
prefareayes	$9,369.513^{***}$
	(1,669.091)
Constant	$-4,038.350$
	(3,409.471)
Observations	546
R^2	0.673
Adjusted R^2	0.666
Note:	*p<0.1; **p<0.05; ***p<0.01

As we can see, lotsize still has a sigificant role, so it is a predictor. However, the coefficient decreased when we introduced new parameters. This is evidence of violation of exogeneity and we can say that the slope is biased. Let's look at the correlations too (I am examining only numeric values for now):

```
kable(cor(Housing[c(1:5, 11)]), booktabs=T,
      caption='Correlation between Numerical Variables') %>%
  kable_styling( latex_options=c('striped', 'hold_position'))
```

We can see that lotsize and price have an acceptable correlation with all other variables. This too is another evidence of OVB.

Table 5: Correlation between Numerical Variables

	price	lotsize	bedrooms	bathrms	stories	garagepl
price	1.0000000	0.5357957	0.3664474	0.5167193	0.4211902	0.3833020
lotsize	0.5357957	1.0000000	0.1518515	0.1938335	0.0836750	0.3528717
bedrooms	0.3664474	0.1518515	1.0000000	0.3737688	0.4079737	0.1391172
bathrms	0.5167193	0.1938335	0.3737688	1.0000000	0.3240656	0.1781783
stories	0.4211902	0.0836750	0.4079737	0.3240656	1.0000000	0.0434119
garagepl	0.3833020	0.3528717	0.1391172	0.1781783	0.0434119	1.0000000

2. *Multivariate Regression*

4. Using the rest of the variables in the dataset, construct a correlation matrix and use it to check if the assumption of exogeneity is valid in the estimated model in question (2). (Explain) Hint: See module 13's Memo 3 for dummies and then see Module 13's Memo 1 to get familiar with OV B, explain accordingly.

ANS.
lm model recognizes factors as different categories and can handle dummy variables, but cor() doesn't understand non-numeric values, so I have to define dummy variables manually for it:

```
Housing.d <- cbind(Housing[1:5],
                   driveway =ifelse(Housing$driveway=='yes', 1, 0),
                   recroom  =ifelse(Housing$recroom =='yes', 1, 0),
                   fullbase =ifelse(Housing$fullbase=='yes', 1, 0),
                   gashw    =ifelse(Housing$gashw   =='yes', 1, 0),
                   airco    =ifelse(Housing$airco   =='yes', 1, 0),
                   Housing[11],
                   prefarea =ifelse(Housing$prefarea=='yes', 1, 0))

kable( head(Housing.d, 2) , booktabs=T, caption='Housing Dataset w/ Dummies') %>%
  kable_styling( latex_options=c('scale_down', 'hold_position', 'striped'))
```

Table 6: Housing Dataset w/ Dummies

price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
42000	5850	3	1	2	1	0	1	0	0	1	0
38500	4000	2	1	1	1	0	0	0	0	0	0

```
kable(cor(Housing.d), booktabs=T, caption='Correlation between All Variables') %>%
  kable_styling( latex_options=c('scale_down', 'striped', 'hold_position'))
```

Many of these variables (Table 7) seem to have relatively high correlation with both price and lotsize. In particular, look at bedrooms, bathrms, driveway, recroom, airco, garagepl and prefarea. So exogeneity was probably violated in the bivariate model.

Table 7: Correlation between All Variables

	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
price	1.0000000	0.5357957	0.3664474	0.5167193	0.4211902	0.2971668	0.2549595	0.1862177	0.0928365	0.4533466	0.3833020	0.3290743
lotsize	0.5357957	1.0000000	0.1518515	0.1938335	0.0836750	0.2887778	0.1403273	0.0474867	-0.0092009	0.2217649	0.3528717	0.2347822
bedrooms	0.3664474	0.1518515	1.0000000	0.3737688	0.4079737	-0.0119963	0.0804923	0.0972006	0.0460278	0.1604122	0.1391172	0.0789527
bathrms	0.5167193	0.1938335	0.3737688	1.0000000	0.3240656	0.0419552	0.1268918	0.1027914	0.0673651	0.1849551	0.1781783	0.0640133
stories	0.4211902	0.0836750	0.4079737	0.3240656	1.0000000	0.1224986	0.0422812	-0.1738596	0.0182605	0.2962157	0.0434119	0.0429397
driveway	0.2971668	0.2887778	-0.0119963	0.0419552	0.1224986	1.0000000	0.0919591	0.0434284	-0.0119421	0.1062901	0.2036822	0.1993781
recroom	0.2549595	0.1403273	0.0804923	0.1268918	0.0422812	0.0919591	1.0000000	0.3724339	-0.0101186	0.1366256	0.0381222	0.1612918
fullbase	0.1862177	0.0474867	0.0972006	0.1027914	-0.1738596	0.0434284	0.3724339	1.0000000	0.0046773	0.0452479	0.0525240	0.2286510
gashw	0.0928365	-0.0092009	0.0460278	0.0673651	0.0182605	-0.0119421	-0.0101186	0.0046773	1.0000000	-0.1303499	0.0681440	-0.0591697
airco	0.4533466	0.2217649	0.1604122	0.1849551	0.2962157	0.1062901	0.1366256	0.0452479	-0.1303499	1.0000000	0.1565956	0.1156259
garagepl	0.3833020	0.3528717	0.1391172	0.1781783	0.0434119	0.2036822	0.0381222	0.0525240	0.0681440	0.1565956	1.0000000	0.0923638
prefarea	0.3290743	0.2347822	0.0789527	0.0640133	0.0429397	0.1993781	0.1612918	0.2286510	-0.0591697	0.1156259	0.0923638	1.0000000

5. Estimate a set of multivariate models to address the potential issue of OVB, adding at most one additional variable each time. (See Memo 2: Multivariate Models under Real World Example) Display all the estimated models side-by-side (you may need two or more stargazer tables here). Using the multivariate models, do you think there is evidence that the estimated parameter in (2) was biased? which of the estimated models you consider the least bias (from now on, we'll call this model the best model)?

Hint: See Module 13's Memo 1 to get familiar with OV B, and follow Memo 2's Multivariate Models to add one variable each time.

ANS.
First, I am not including gashw in the model, because it has very poor correlation with both price and lotsize. Let's check other variables one by one:

```
mv.model.1 <- lm(price ~ lotsize + bedrooms, data=Housing.d)
mv.model.2 <- lm(price ~ lotsize + bedrooms + bathrms, data=Housing.d)
mv.model.3 <- lm(price ~ lotsize + bedrooms + bathrms + driveway,
                 data=Housing.d)
mv.model.4 <- lm(price ~ lotsize + bedrooms + bathrms + driveway + recroom,
                 data=Housing.d)
mv.model.5 <- lm(price ~ lotsize + bedrooms + bathrms + driveway + recroom +
                 airco, data=Housing.d)
mv.model.6 <- lm(price ~ lotsize + bedrooms + bathrms + driveway + recroom +
                 airco + garagepl, data=Housing.d)
mv.model.7 <- lm(price ~ lotsize + bedrooms + bathrms + driveway + recroom +
                 airco + garagepl + prefarea, data=Housing.d)
mv.model.8 <- lm(price ~ lotsize + bedrooms + bathrms + driveway + recroom +
                 airco + garagepl + prefarea + fullbase, data=Housing.d)
mv.model.9 <- lm(price ~ lotsize + bedrooms + bathrms + driveway + recroom +
                 airco + garagepl + prefarea + fullbase + stories, data=Housing.d)


stargazer(list(bivar.model, mv.model.1, mv.model.2, mv.model.3, mv.model.4),
          align=T, no.space=T, header=F,
          title="Comparison of Models Part 1",
          column.sep.width = "1pt",
          dep.var.labels=c("Price"),
          omit.stat=c("LL", "ser", "f"),
          table.placement = "H")
```

Table 8: Comparison of Models Part 1

	Dependent variable:				
	Price				
	(1)	(2)	(3)	(4)	(5)
lotsize	6.599***	6.053***	5.411***	4.785***	4.626***
	(0.446)	(0.424)	(0.388)	(0.395)	(0.391)
bedrooms		10,567.350***	5,826.802***	6,196.547***	6,052.177***
		(1,247.676)	(1,206.571)	(1,177.634)	(1,159.817)
bathrms			19,750.210***	19,688.790***	19,052.260***
			(1,785.083)	(1,739.429)	(1,718.864)
driveway				13,144.810***	12,560.470***
				(2,405.969)	(2,372.514)
recroom					8,952.255***
					(2,097.103)
Constant	34,136.190***	5,612.600	−2,418.293	−11,501.410***	−10,526.990***
	(2,491.064)	(4,102.819)	(3,779.412)	(4,040.559)	(3,984.282)
Observations	546	546	546	546	546
R^2	0.287	0.370	0.486	0.513	0.529
Adjusted R^2	0.286	0.368	0.483	0.510	0.525
Note:					*p<0.1; **p<0.05; ***p<0.01

```
stargazer(list(mv.model.5, mv.model.6, mv.model.7, mv.model.8, mv.model.9),
          align=T, no.space=T, header=F,
          title="Comparison of Models Part 2",
          column.sep.width = "1pt",
          dep.var.labels=c("Price"),
          omit.stat=c("LL", "ser", "f"),
          table.placement = "H")
```

Table 9: Comparison of Models Part 2

	Dependent variable:				
	Price				
	(1)	(2)	(3)	(4)	(5)
lotsize	4.101***	3.639***	3.316***	3.349***	3.536***
	(0.368)	(0.376)	(0.370)	(0.370)	(0.355)
bedrooms	5,194.116***	4,924.599***	4,671.830***	4,597.049***	1,919.541*
	(1,082.922)	(1,066.079)	(1,037.370)	(1,037.494)	(1,061.250)
bathrms	17,543.270***	16,912.280***	17,013.920***	16,922.830***	14,677.400***
	(1,607.321)	(1,586.089)	(1,542.058)	(1,541.634)	(1,508.028)
driveway	11,581.140***	10,422.160***	8,759.434***	8,800.531***	6,638.478***
	(2,209.577)	(2,187.170)	(2,146.379)	(2,144.239)	(2,073.499)
recroom	7,308.760***	7,645.728***	6,364.557***	5,336.898***	4,519.340**
	(1,958.949)	(1,926.867)	(1,886.790)	(2,010.065)	(1,926.238)
airco	15,202.320***	14,748.280***	14,349.720***	14,420.930***	11,693.250***
	(1,648.742)	(1,623.683)	(1,580.061)	(1,579.094)	(1,558.328)
garagepl		4,070.540***	4,121.579***	4,071.954***	4,512.089***
		(911.278)	(885.966)	(885.651)	(849.458)
prefarea			9,854.542***	9,363.314***	9,007.644***
			(1,735.582)	(1,765.574)	(1,689.676)
fullbase				2,378.280	5,558.221***
				(1,616.814)	(1,609.766)
stories					6,678.946***
					(937.576)
Constant	−7,017.655*	−4,769.932	−3,049.459	−3,436.989	−4,115.163
	(3,725.855)	(3,696.439)	(3,606.332)	(3,612.051)	(3,456.578)
Observations	546	546	546	546	546
R^2	0.593	0.608	0.630	0.631	0.663
Adjusted R^2	0.589	0.603	0.624	0.625	0.657
Note:					*p<0.1; **p<0.05; ***p<0.01

Now let's talk about the models:

1. As we add more variables, the amplitude of β for lotsize decreases, this tells us that there was OVB. However, this variable (lotsize), remains a strong and significant predictor even after adding new independent variables. During the addition of extra parameters we can see that addition of recroom had a very small effect on the coefficient of lotsize. This means that we can probably remove it.

2. Let's look at how adjusted R^2 increased after addition of variables:

- mv.1: bedrooms: ~ + %8.2
- mv.2: bathrms: ~ + %11.5
- mv.3: driveway: ~ + %2.7
- mv.4: recroom: ~ + %1.5
- mv.5: airco: ~ + %6.4
- mv.6: garagepl: ~ + %1.4
- mv.7: prefarea: ~ + %2.1
- mv.8: fullbase: ~ + %0.1
- mv.9: stories: ~ + %3.2
  We can see that fullbase almost didn't change the adjusted R^2 at all.

3. By looking at the significance level of the variables in the last model (mv.9) we can also see that bedrooms and recroom are not significant any more.

So let's now look at a model with these variables: lotsize, bathrooms, driveway, airco, garagepl, prefarea and stories:

```
best.mv.model <- lm(price ~ lotsize + bathrms + driveway + airco +
                    garagepl + prefarea + stories, data=Housing.d)

stargazer(best.mv.model,
          align=T, no.space=T, header=F,
          title="Best MV Model",
          column.sep.width = "1pt",
          dep.var.labels=c("Price"),
          omit.stat=c("LL", "ser", "f", "n"),
          table.placement = "H")
```

Table 10: Best MV Model

	Dependent variable:
	Price
lotsize	3.589***
	(0.362)
bathrms	16,577.790***
	(1,487.454)
driveway	6,613.206***
	(2,111.809)
airco	12,446.700***
	(1,591.726)
garagepl	4,592.153***
	(868.692)
prefarea	11,072.200***
	(1,689.155)
stories	6,364.343***
	(872.385)
Constant	1,416.563
	(2,795.540)
R^2	0.643
Adjusted R^2	0.638
Note:	*p<0.1; **p<0.05; ***p<0.01

This model looks very good. All of the variables are significant, the model doesn't have too many vaiables and has a good adjusted R^2. I'll keep this configuration as the best multivariate model.

6. Check if the best model suffers from multicollinearity (if it does, don't try to fix it, just explain β what problems it may cause).

Hint:Use vif() in car package to easily calculate VIF and lecture 13's Memo 2 of multicollinearity to explain.

ANS.

```
# Looking at multicollinearity:
kable(t(vif(best.mv.model)), booktabs=T, caption='Best Model Multicollinearity Test') %>%
  kable_styling( latex_options=c('hold_position'))
```

Table 11: Best Model Multicollinearity Test

lotsize	bathrms	driveway	airco	garagepl	prefarea	stories
1.3011	1.17943	1.144163	1.161466	1.183454	1.084527	1.212726

As we can see in Table 11, all values are below 5. This means weak imperfect multicollinearity and is not an issue at all. We can also test the model with all of the variabes:

```
kable(t(vif(mv.model.9)), booktabs=T, caption='All Vars Model Multicollinearity Test') %>%
  kable_styling( latex_options=c('hold_position', 'scale_down'))
```

Table 12: All Vars Model Multicollinearity Test

lotsize	bedrooms	bathrms	driveway	recroom	airco	garagepl	prefarea	fullbase	stories
1.321558	1.365033	1.278248	1.163049	1.2105	1.173814	1.193205	1.144247	1.316175	1.476968

As we can see in Table 12, here too, the multicollinearity isn't an issue at all. All of the values are below 5, which means weak imperfect multicollinearity.

3. *Non – linear Functional Forms*

7. Take a look at the graph from part (1), do you think there is any reason to believe that the effect of lot size on price is not the same for all the domain of lot size? if yes, is the effect increasing or decreasing?

ANS.
Looking closer at the data, we can see that the effect of the lotsize is fairly linear with a positive slope up to 4000~5000 sq.ft. At some point around 6000 sq.ft, it looks like the slope is either decreased (the prices are stagnated) or the prices have a sudden jump. After that we really don't have a lot of data points to have a fair judgement about the trend of the data but it seems like there is no more steep slope and this means that at higher lotsizes, the prices are not increasing sharply with lotsizes and we can say that while the slope is sharp at the start, it gets flatter after a point in higher lotsizes.

8. Estimate the best model again, but this time transform the lot size variable to natural logarithms. Interpret the estimated parameter for log of the lot size.

ANS.

```
best.mv.log <- lm(price ~ I(log(lotsize)) + bathrms + driveway + airco +
                  garagepl + prefarea + stories, data=Housing.d)

stargazer(best.mv.log, align=T, no.space=T, header=F,
          title="Best MV model + log term",
          column.sep.width = "1pt",
          dep.var.labels=c("Price"),
          omit.stat=c("LL", "ser"),
          table.placement = "H")
```

Table 13: Best MV model + log term

	Dependent variable:
	Price
I(log(lotsize))	$20,629.980^{***}$
	(1,999.470)
bathrms	$16,488.610^{***}$
	(1,478.482)
driveway	$5,332.450^{**}$
	(2,123.777)
airco	$11,673.250^{***}$
	(1,592.635)
garagepl	$4,511.528^{***}$
	(863.299)
prefarea	$11,468.190^{***}$
	(1,671.365)
stories	$6,288.332^{***}$
	(866.665)
Constant	$-153,202.800^{***}$
	(16,171.270)
Observations	546
R^2	0.648
Adjusted R^2	0.643
F Statistic	141.201^{***} (df = 7; 538)
Note:	*p<0.1; **p<0.05; ***p<0.01

The estimated coefficient for the $log(lotsize)$ term is 20,629.980. Let's see what it means. Let's say we increase the lotsize of a property 1 square ft ($x \rightarrow x+1$), how much does the price change?
$price_1 = 20,629.980 \times log(x)$
$price_2 = 20,629.980 \times log(x+1)$
$price_2 - price_1 = 20,629.980(log(x+1) - log(x)) = 20,629.980 \times log(\frac{x+1}{x})$
This means that a unit square ft of increase in the lotsize from x $sq.ft$ to $(x+1)$ $sq.ft$, increase/deacreases the price as much as $20,629.980$ $log(\frac{x+1}{x})$. Note that we didn't include other parameters in our calculations because if they remain constant, they will be cancelled out of the equations.

9. Estimate the best model twice: (a) first, adding a quadratic term for lot size, and, (b) second, adding a quadratic and cubic terms. Using the change in lot size as a one standard deviation change from the mean, compare the effect of lot size in the original model, model (a), and, model (b). Can you reject the hypothesis that the relation between lot size and price is linear? quadratic? cubic? (Explain)

ANS.

```
# a) lotsize quadratic:
best.mv.quad <- lm(price ~ lotsize + I(lotsize^2) + bathrms + driveway + airco +
                   garagepl + prefarea + stories, data=Housing.d)

stargazer(best.mv.quad, align=T, no.space=T, header=F,
          title="Best MV model + quadratic term",
          column.sep.width = "1pt",
```

```
          dep.var.labels=c("Price"),
          omit.stat=c("LL", "ser"),
          table.placement = "H")
```

Table 14: Best MV model + quadratic term

	Dependent variable:
	Price
lotsize	6.345***
	(1.254)
I(lotsizeˆ2)	−0.0002**
	(0.0001)
bathrms	16,505.590***
	(1,481.930)
driveway	5,843.841***
	(2,130.057)
airco	11,850.100***
	(1,606.643)
garagepl	4,441.388***
	(867.763)
prefarea	10,971.870***
	(1,683.070)
stories	6,246.720***
	(870.460)
Constant	−5,184.110
	(4,003.984)
Observations	546
R^2	0.647
Adjusted R^2	0.641
F Statistic	122.787*** (df = 8; 537)
Note:	*p<0.1; **p<0.05; ***p<0.01

```
# b) lotsize quadratic and cubic:
best.mv.quad.cub <- lm(price ~ lotsize + I(lotsizeˆ2) + I(lotsizeˆ3) + bathrms +
                       driveway + airco +garagepl + prefarea + stories, data=Housing.d)

stargazer(best.mv.quad.cub, align=T, no.space=T, header=F,
          title="Best MV model + quadratic and cubic term",
          column.sep.width = "1pt",
          dep.var.labels=c("Price"),
          omit.stat=c("LL", "ser"),
          table.placement = "H")
```

Table 15: Best MV model + quadratic and cubic term

	Dependent variable:
	Price
lotsize	13.402***
	(3.661)
I(lotsizeˆ2)	−0.001**
	(0.001)
I(lotsizeˆ3)	0.00000**
	(0.00000)
bathrms	16,345.000***
	(1,479.599)
driveway	5,209.573**
	(2,146.126)
airco	11,583.830***
	(1,607.120)
garagepl	4,636.680***
	(870.408)
prefarea	11,650.670***
	(1,710.398)
stories	6,302.321***
	(868.297)
Constant	−18,274.240**
	(7,528.216)
Observations	546
R^2	0.649
Adjusted R^2	0.643
F Statistic	110.263*** (df = 9; 536)
Note:	*p<0.1; **p<0.05; ***p<0.01

```
# Making the dataset to feed into predict()
mean <- mean(Housing.d$lotsize)
mean.plus.sd <- mean + sd(Housing.d$lotsize)

check.data <- Housing.d[1:2, 2:12]
check.data[1:11] <- 1
check.data[1:2, 1] <- c(mean, mean.plus.sd)
# ^ Note that since we are examining lotsize, it was only important to have the values
# for lotsize, all other values are equal to 1 in both rows:

kable( head(check.data, 2) , booktabs=T, caption='Feeded data') %>%
kable_styling( latex_options=c('scale_down', 'hold_position', 'striped'))
```

Table 16: Feeded data

lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
5150.266	1	1	1	1	1	1	1	1	1	1
7318.424	1	1	1	1	1	1	1	1	1	1

```r
# Change in price by changing lotsize from mean(lotsize) to mean + sd(lotsize):

# a) Multivariate model (linear lotsize)
linear.preds <- predict(best.mv.model, check.data)
linear.diff  <- linear.preds[2] - linear.preds[1]

# b) Multivariate model (quadratic lotsize)
quad.preds <- predict(best.mv.quad, check.data)
quad.diff  <- quad.preds[2] - quad.preds[1]

# c) Multivariate model (quadratic+cubic lotsize)
quad.cub.preds <- predict(best.mv.quad.cub, check.data)
quad.cub.diff  <- quad.cub.preds[2] - quad.cub.preds[1]

# Dispalying the results:
diff.results <- data.frame(linear.model=linear.diff,
                           quadratic.model=quad.diff,
                           quadratic.cubic.model=quad.cub.diff)
row.names(diff.results) <- NULL

kable(head(diff.results), booktabs=T,
      caption='Comparison of Differences for Different Models') %>%
  kable_styling( latex_options=c('hold_position'))
```

Table 17: Comparison of Differences for Different Models

linear.model	quadratic.model	quadratic.cubic.model
7782.569	8294.893	6648.549

Let's look at the plot in question 1 again, but this time let's highlight our regions of interest:

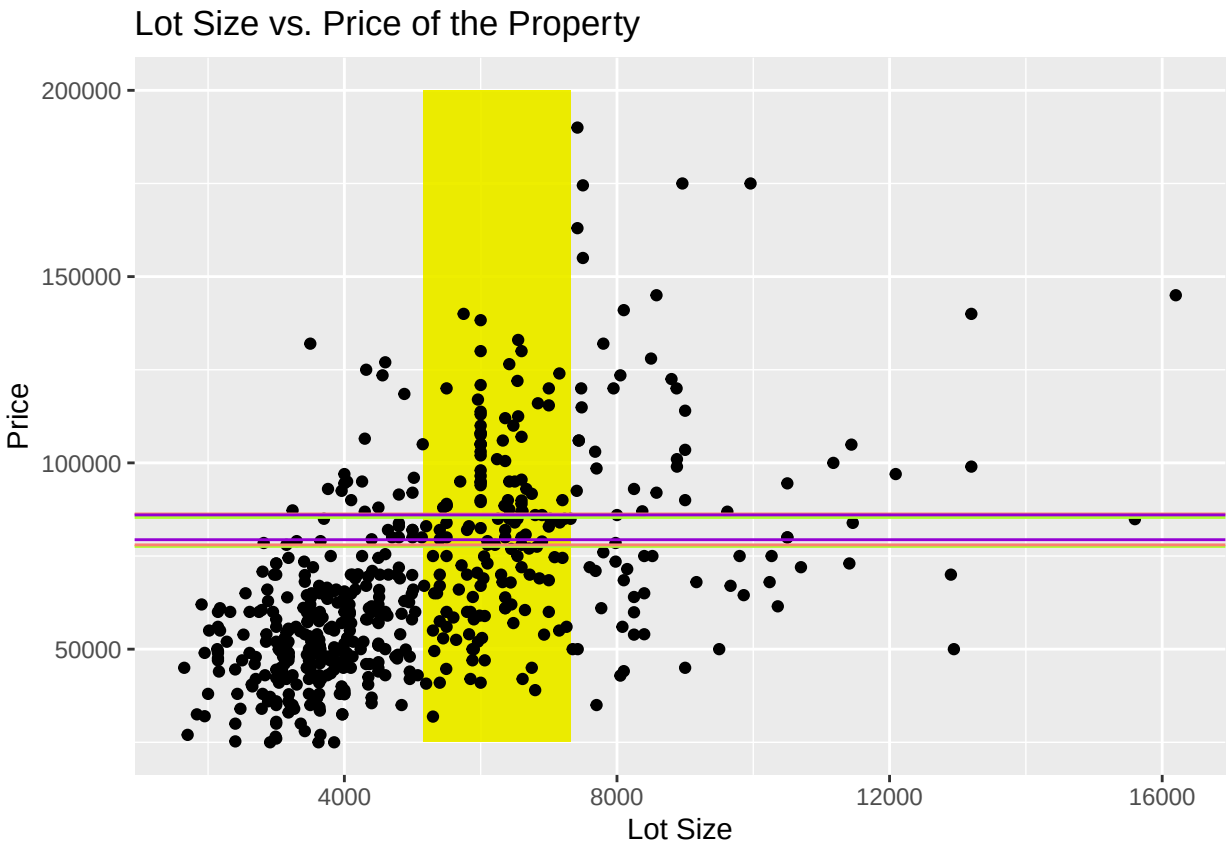
```r
ggplot(data=Housing, aes(x=lotsize, y=price)) + geom_point() +
  geom_rect(aes(xmin=5150, xmax=7318, ymin=25000,
                ymax=200000), colour="transparent", fill="yellow2", alpha=0.005) +

  geom_hline(color='greenyellow', yintercept=linear.preds[1]) +
  geom_hline(color='greenyellow', yintercept=linear.preds[2]) +

  geom_hline(color='salmon', yintercept=quad.preds[1]) +
  geom_hline(color='salmon', yintercept=quad.preds[2]) +

  geom_hline(color='darkviolet', yintercept=quad.cub.preds[1]) +
  geom_hline(color='darkviolet', yintercept=quad.cub.preds[2]) +

  xlab('Lot Size') + ylab('Price') +
  ggtitle('Lot Size vs. Price of the Property')
```



The yellow region is from mean of lotsize to one standard deviation after the mean. I have added lines showing the location of predicted prices by each model. The green line shows the predictions of the linear model; the pink lines show the predictions of the quadratic model, and the dark violet lines show the predictions of the cubic model.

This is honestly a tricky question. It looks like the correct answer for this question shouldn't be the linear model but the evidence that I am seeing tells me otherwise. At least, there is little benefit in using cubic model. Let's look at some of our findings:

- For linear(lotsize) model:
  Coeff (lotsize) = 3.599 → p-value = < 2e-16 (significant)
  Residual standard error: 15830 on 537 degrees of freedom
  Multiple R-squared: 0.6535, Adjusted R-squared: 0.6484
  p-value: < 2.2e-16
- For quadratic(lotsize) model:
  Coeff (lotsize) = 6.345 → p-value = 5.77e-07 (significant)
  Coeff (lotsize^2) = -2.020e-04 → p-value = 0.02217 (Ok)
  Residual standard error: 15990 on 537 degrees of freedom
  Multiple R-squared: 0.6465, Adjusted R-squared: 0.6413
  p-value: < 2.2e-16
- For cubic(lotsize) model:
  Coeff (lotsize) = 13.40 → p-value = 0.000277 (significant)
  Coeff (lotsize^2) = -1.253e-03 → p-value = 0.016283 (Better)
  Coeff (lotsize^3) = 4.487e-08 → p-value = 0.040761 (Ok?)
  Residual standard error: 15950 on 536 degrees of freedom
  Multiple R-squared: 0.6493, Adjusted R-squared: 0.6434

p-value: $<$ 2.2e-16

So, As it can be seen, the error of the models have increased as we added non-linear terms, we have made the models more complex with parameters which are not super significant, and we have gained very little increase in adjusted R^2 from quadratic to cubic and compared to the linear model both of these are even lower! So, I would ask why bother doing it at all? Let's look at another piece of evidence here, R^2 has increased, but the adjusted R^2 has decreased in the nonlinear models. This itself is telling us that the addition of the new non-linear terms are not helping the model.

On the other hand, there is only one argument that may justify using the cubic model. If based on the scatter plot, we argue that there is a sharp slope in the beginning, a stagnated region with flat slope afterwards, and a much less sharp slope region after that at the end, this behavior will need a cubic function to be explained, and the fact that $price_{mean+sd(lotsize)} - price_{mean(lotsize)}$ for the cubic model is less than the quadratic and linear model, shows that it has better captured the stagnated central region.

10. Using the best model as the nested model, test the hypothesis that the effect of lot size on price is moderated by prefarea.

ANS.
In order to check for the moderation effect between lotsize and prefarea, I'll add an interaction term lotsize×prefarea to the model and check if it is significant.

```
mod.effect <- lm(price ~ lotsize + bathrms + driveway + airco +
                 garagepl + prefarea + stories + lotsize*prefarea, data=Housing.d)

stargazer(mod.effect, align=T, no.space=T, header=F,
          title="Best MV model with Interaction Term",
          column.sep.width = "1pt",
          dep.var.labels=c("Price"),
          omit.stat=c("LL", "ser"),
          table.placement = "H")
```

Table 18: Best MV model with Interaction Term

	Dependent variable:
	Price
lotsize	3.238***
	(0.424)
bathrms	16,640.530***
	(1,485.861)
driveway	7,121.384***
	(2,132.833)
airco	12,393.790***
	(1,589.809)
garagepl	4,494.461***
	(869.625)
prefarea	4,421.537
	(4,505.717)
stories	6,435.765***
	(872.299)
lotsize:prefarea	1.154
	(0.725)
Constant	2,583.945
	(2,886.289)
Observations	546
R^2	0.645
Adjusted R^2	0.639
F Statistic	121.831*** (df = 8; 537)
Note:	*p<0.1; **p<0.05; ***p<0.01

Judging by the p-value of 0.112018 for the interaction term (lotsize×prefarea), which is not significant I can say there is no moderation effect here.

4. *Unsupervised Machine Learning*

11. Run a factor analysis or PCA on the Housing dataset, examine the loadings of the factors on the variables. Sort the variables by their loadings, and try to interpret what the first one "mean".

ANS.
I am using PCA. Since PCA is a variance-based model, it is necessary to scale all variables to be in the same order first.

Also, I am removing price from the data because I think it is the dependent variable and everything else are independent variables which predict price. Therefore, it makes sense to take it out of the data before running PCA.

```
# Scaling the dataset between 0 to 1:
Housing.d.sc <- apply(Housing.d[2:12], 2,
                      FUN=function(x){(x-min(x))/(max(x)-min(x))})
                      # the function scales between 0 to 1

kable(head(Housing.d.sc, 2), booktabs=T, caption='Housing Dataset Scaled w/ Dummies') %>%
  kable_styling( latex_options=c('scale_down', 'hold_position', 'striped'))
```

Table 19: Housing Dataset Scaled w/ Dummies

lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
0.2886598	0.4	0	0.3333333	1	0	1	0	0	0.3333333	0
0.1615120	0.2	0	0.0000000	1	0	0	0	0	0.0000000	0

```
# PCA using singular value decomposition (SVD):
pca <- prcomp(Housing.d.sc)
# 1st component:
kable(t(pca$rotation[ ,1][order(pca$rotation[ ,1])]), booktabs=T,
      caption='First Principal Component') %>%
  kable_styling( latex_options=c('hold_position', 'scale_down')) # order() for sorting
```

Table 20: First Principal Component

gashw	stories	bedrooms	bathrms	lotsize	garagepl	driveway	airco	recroom	prefarea	fullbase
-0.024903	0.0382487	0.0510168	0.0670522	0.0790251	0.1093195	0.1753718	0.3846337	0.4184918	0.4404984	0.6517284

The result is interesting. Here we can see the two poles of the 1st PC. On one side, we have stories, number of bedrooms and bathrooms, lotsize and generally variables which are related to the size of the property (with the exception of gashw). On the other side, we can see variables mostly about amenities of the building like whether it has a full finished basement or does the house has recreational room or an airconditioner or is it in a preferred neighborhood?

I think one side of the first PC (Table 20) is where most of the home buyer will consider: the main/basic features, the other side is probably the concern of wealthier buyers which are looking for more than the basic features of the property.

Let's do a factor analysis too:

```
# Factor Analysis:
fact <- fa(Housing.d.sc, nfactors=2)
# 1st component:
kable(t(fact$loadings[ ,1][order(fact$loadings[ ,1])]), booktabs=T,
      caption='First Factor') %>%
  kable_styling( latex_options=c('hold_position', 'scale_down')) # order() for sorting
```

Table 21: First Factor

fullbase	recroom	prefarea	driveway	gashw	garagepl	lotsize	airco	bathrms	bedrooms	stories
-0.2056258	-0.0480228	-0.0429031	0.0399821	0.0405236	0.12022	0.1279492	0.2969513	0.4412643	0.5095179	0.7778175

The result here is very similar to PCA and maybe even better! Looking at Table 21, on the left side we can see variables like fullbase, recroom, prefarea (aminities/secondary factors), on the other side there are stories, bedroom, bathroom, ... (size of the property/fundamental factors).

12. Use k-means algorithm and examine the centers of each cluster using only two centroids. How are they similar to and different from the factor loadings of the first factor?

ANS.

```r
set.seed(1)
# Running kmeans w/ 2 centers and 25 random start:
kmeans <- kmeans(Housing.d.sc, centers=2, nstart=25)

kmeans.centroids <- kmeans$centers

kmeans.topvars_centroid1 <- kmeans.centroids[1, order(kmeans.centroids[1, ])]
kable(t(tail(kmeans.topvars_centroid1)), booktabs=T, caption='Fisrt Centroid') %>%
  kable_styling( latex_options=c('hold_position'))
```

Table 22: Fisrt Centroid

airco	prefarea	recroom	bedrooms	driveway	fullbase
0.3455497	0.3664921	0.3717277	0.4125654	0.8795812	1

```r
kmeans.topvars_centroid2 <- kmeans.centroids[2, order(kmeans.centroids[2, ])]
kable(t(tail(kmeans.topvars_centroid2)), booktabs=T, caption='Second Centroid') %>%
  kable_styling( latex_options=c('hold_position'))
```

Table 23: Second Centroid

garagepl	lotsize	airco	stories	bedrooms	driveway
0.2197183	0.2353824	0.3014085	0.3061033	0.3825352	0.8478873

Here, the first centroid (Table 22), has more in common with the right pole of the first PC, the amenities and the secondary features (with the exception of the bedrooms). The second centroid (Table 23), has more in with the left pole of the PC, the first order of importance variables, the more basic and fundamental features mainly about the size of the building.

5. *Supervised Machine Learning*

13. Divide the Housing data into two equally sized samples (one for training and one for testing). The dependent variable is price. Using the training sample, estimate a ridge model using the Housing dataset and find the optimal value of λ.

ANS.

```r
# Selecting test/train samples:
# Scaling data:
Housing.d.sc <- apply(Housing.d, 2,
                      FUN=function(x){(x-min(x))/(max(x)-min(x))})

set.seed(1)
train.ind <- sample(1:nrow(Housing.d.sc), nrow(Housing.d.sc)/2)

data.x  <- Housing.d.sc[ ,2:12]
data.y  <- Housing.d.sc[ ,1]

train.x <- data.x[train.ind, ]
```

```r
train.y <- data.y[train.ind]

test.x  <- data.x[-train.ind, ]
test.y  <- data.y[-train.ind]

lambdas <- 10^seq(7, -2, length=100) # 100 values from 10^-2 to 10^7

# Using cv.glmnet to find the best value of lambda:
# alpha = 0: Purely Ridge model
ridge.model <- cv.glmnet(train.x, train.y, alpha=0, lambda=lambdas)
# Choosing best lambda:
best.lambda <- ridge.model$lambda.min
cat('Optimal value of Lambda for Ridge model is: ', best.lambda)
```

```
## Optimal value of Lambda for Ridge model is:  0.02848036
```

14. How does the model performs in the testing sample? Compare the results of the ridge model with a linear regression. Which model performs best?

ANS.
It is not exactly stated which linear model. So, I will compare it with one bivariate model (price ~ lotsize) and one multiple regression model with all variables:

```r
# Checking the performance of Ridge model on test data:
test.y.prediction.r <- predict(ridge.model$glmnet.fit, s=best.lambda, newx=test.x)
mse.test.r <- sum((test.y - test.y.prediction.r)^2) / nrow(test.x)
mse.test.r
```

```
## [1] 0.00750913
```

```r
# Linear regression model:
# a. Biavriate model:
bivar.model <- lm(train.y ~ train.x[ ,1]) # price ~ lotsize
test.y.prediction.b <- cbind(1, test.x[ ,1]) %*% bivar.model$coefficients
mse.test.b <- sum((test.y - test.y.prediction.b)^2) / nrow(test.x)
mse.test.b
```

```
## [1] 0.01518691
```

```r
# b. Multivariate model:
multivar.model <- lm(train.y ~ train.x) # price ~ all variables
test.y.prediction.m <- cbind(1, test.x) %*% multivar.model$coefficients
mse.test.m <- sum((test.y - test.y.prediction.m)^2) / nrow(test.x)
mse.test.m
```

```
## [1] 0.007563338
```

As we can see, the bivariate model has a ~ %101.1 larger error compared to th Ridge model but the multivariate model is very close to Ridge model with only ~ %2.3 larger error. Overall, Ridge model did a better job on test data.

15. Using the HealthInsurance dataset. Divide the data into two equally sized samples (one for training and one for testing). The dependent variable is health. Using the training sample; and a radial kernel and the following two values for cost C = (1e - 05, 1e + 01), estimate a support vector machine model and choose the optimal cost parameter using the function tune. (In this part, feel free to reduce the size of each sample to improve the speed of the calculations.)

ANS.

```
# 1. Preparing data for svm:
HealthInsurance.d <- HealthInsurance
# Making variables dummy:
HealthInsurance.d[ ,c(1, 3, 5:7)] <- apply(HealthInsurance.d[ ,c(1, 3, 5:7)], 2,
                                           FUN=function(x){ifelse(x=='yes', 1, 0)})
HealthInsurance.d$gender <- ifelse(HealthInsurance.d$gender=='male', 0, 1)

# A function for making dummy vars (drops the 1st lvl)
dummy.creator <- function(x){
  level <- levels(x)[-1] # droping the first level
  y <- data.frame(ifelse(x==level[1], 1, 0))
  for (i in 2:length(level)) {
    z <- data.frame(ifelse(x==level[i], 1, 0))
    y <- cbind(y, z)
  }
  colnames(y) <- level
  return(y)
}

HealthInsurance.d <- cbind(HealthInsurance.d[ ,1:8],
                           dummy.creator(HealthInsurance.d[ ,9]),
                           dummy.creator(HealthInsurance.d[ ,10]),
                           dummy.creator(HealthInsurance.d[ ,11]))
# Scaling between 0 and 1:
HealthInsurance.d.sc <- data.frame(apply(HealthInsurance.d, 2,
                                         FUN=function(x){(x-min(x))/(max(x)-min(x))}))
# Making the dependent variable dummy:
HealthInsurance.d.sc$health <- as.factor(HealthInsurance.d.sc$health)

kable( head(HealthInsurance.d.sc, 2) , booktabs=T,
       caption='Health Insurance Dataset Scaled w/ Dummies') %>%
  kable_styling( latex_options=c('scale_down', 'hold_position', 'striped'))
```

Table 24: Health Insurance Dataset Scaled w/ Dummies

health	age	limit	gender	insurance	married	selfemp	family	midwest	south	west	afam	cauc	ged	highschool	bachelor	master	phd	other
1	0.2954545	0	0	1	1	1	0.2307692	0	1	0	0	1	0	0	1	0	0	0
1	0.2954545	0	1	1	1	0	0.2307692	0	1	0	0	1	0	1	0	0	0	0

```
# 2. Selecting the test/train samples:
set.seed(1)
train.ind  <- sample(1:nrow(HealthInsurance.d.sc), nrow(HealthInsurance.d.sc)/2)

train.data <- HealthInsurance.d.sc[train.ind, ]
test.data  <- HealthInsurance.d.sc[-train.ind, ]
```

```r
# 3. Choosing the best SVM model:
cost.values <- c(1e-5, 1e+1)

svm <- tune(svm, health~., data=train.data,
            ranges=list(cost=cost.values), kernel="radial")
summary(svm)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##   cost
##  1e-05
##
## - best performance: 0.06861987
##
## - Detailed performance results:
##    cost      error dispersion
## 1 1e-05 0.06861987 0.01370654
## 2 1e+01 0.07657390 0.01286117
```

16. How does the svm model performs in the testing sample? How does the model compares to a logit in terms of accuracy?

ANS.

```r
# 1. Performance of the SVM model:
svm.pred.test.y <- predict(svm$best.model, newdata=test.data)

# Percentage of correct predictions using the test data:
sum(svm.pred.test.y==test.data$health)*100/nrow(test.data)
```

```
## [1] 92.56987
```

```r
# 2. Logit model:
logit <- glm(health~., data=train.data, family="binomial")

# 3. Performance of Logit model:
logit.ped.test.y <- round( predict(logit, newdata=test.data, type="response") )
# The "reponse" option was used to the get predicted probabilities,
# and then the results were rounded, so that any predicted prob > 0.5 is a 1,
# and vice versa for 0.

# Percentage of correct predictions:
sum(logit.ped.test.y==test.data$health)*100/nrow(test.data)
```

```
## [1] 92.52443
```

As we can see, the two models have almost similar performance. Logit model did a hair better but they are so close that we can't really announce a winner between two models.

However, there is one thing that I think needs to be discussed. The SVM model, only returns Yes for health status, meaning it predicts all of the subjects as healthy. You might think why a model that always predicts one thing has such accuracy? It is because the data is unbalanced. Let's take a closer look at the data:

```r
# Percentage of Yes in the health status column:
nrow(HealthInsurance.d.sc[HealthInsurance.d.sc$health==1, ])*100/
  nrow(HealthInsurance.d.sc)
```

```
## [1] 92.8539
```

As we can see, %92.8 of the people in the dataset are healthy, so any random model which always predicts 'Yes' for health status, will get a %92.8 accuracy. So, just by looking at the correct prediction rate we can't be really sure if the SVM model actually did a good job or not. There are special procedures to deal with these types of unbalanced datasets which I assume weren't the purpose of this question and I would suffice to just mention what is happening here.