DATA 303 Assignment 2

Izzy Southon, 300597453

Due 11:59pm Friday 19 April

Assignment Questions

Q1.(26 marks) The dataset `fiat.csv`contains 1538 records on sales of used Fiat 500 cars in Italy. The variables in the dataset are:

- `model`: Fiat 500 comes in several 'flavours' :'pop', 'lounge', 'sport'
- `power`: number of KW of the engine
- `age`: age of the car in number of days (at the time dataset was created)
- `km`: Distance travelled by the car in kilometers
- `owners`: number of previous owners
- `lat`: latitude of the seller (the price of cars in Italy varies from North to South of the country)
- `lon`: longitude of the seller
- `price`: selling price (in Euro)

In this question, we are interested in identifying the key predictors of `price`, and in understanding how these predictors affect `price`. Model interpretability is important in this case. The initial steps in building a model for `price` are shown in the Appendix on pages 3 to 6.

As there is evidence of non-normality and non-constant variance, a log-transformation for `price` is to be applied in the rest of the analyses. Prepare the data as has been done in the Appendix, and use your new dataset to answer the questions below.

```
library(dplyr)
library(memisc)
```

```
fiat<-read.csv("fiat.csv", header=T)
str(fiat)
```

```
## 'data.frame':    1538 obs. of  8 variables:
##  $ model : chr  "lounge" "pop" "sport" "lounge" ...
##  $ power : int  51 51 74 51 73 74 51 51 73 51 ...
##  $ age   : int  882 1186 4658 2739 3074 3623 731 1521 4049 3653 ...
##  $ km    : int  25000 32500 142228 160000 106880 70225 11600 49076 76000 89000 ...
##  $ owners: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ lat   : num  44.9 45.7 45.5 40.6 41.9 ...
##  $ lon   : num  8.61 12.24 11.42 17.63 12.5 ...
##  $ price : int  8900 8800 4200 6000 5700 7900 10750 9190 5600 6000 ...
```

```
## Changing owners and power into categorical variables as they have very few unique values
fiat$owners<-as.factor(fiat$owners)
fiat<-fiat%>%
  mutate(power.cat=memisc::recode(power,"50-59"<-c(50:59),
                                  "60-69"<-c(60:69),
                                  "70-79"<-c(70:79)))%>%
  dplyr::select(-power)
str(fiat)
```

```
## 'data.frame':    1538 obs. of  8 variables:
##  $ model    : chr  "lounge" "pop" "sport" "lounge" ...
##  $ age      : int  882 1186 4658 2739 3074 3623 731 1521 4049 3653 ...
##  $ km       : int  25000 32500 142228 160000 106880 70225 11600 49076 76000 89000 ...
##  $ owners   : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  $ lat      : num  44.9 45.7 45.5 40.6 41.9 ...
##  $ lon      : num  8.61 12.24 11.42 17.63 12.5 ...
##  $ price    : int  8900 8800 4200 6000 5700 7900 10750 9190 5600 6000 ...
##  $ power.cat: Factor w/ 3 levels "50-59","60-69",..: 1 1 3 1 3 3 1 1 3 1 ...
```

a. **(4 marks)** Fit a generalised additive model for `log(price)`, including all predictors used in `fit1` in the Appendix. Use a smooth spline for each numerical predictor. Comment on the non-linearity and significance of all smooth terms.

$Y = \beta_0 + s_1(X_1) + s_2(X_2) + ... + s_p(X_p) + \epsilon$

The fitted model equation where Y = log(price) is:

$log(price) = 9.0346 + 3.78(age) + 2.92(km) + 4.54(lat) + 4.34(lon) + \epsilon$

```
library(mgcv)
library(pander)
```

```
# Multiple regression with GAMs - GAMs with multiple predictors

# Fitting the GAM model for `log(price)`
fit.gam<-gam(log(price)~model + owners + power.cat + s(age) + s(km) + s(lat)
             + s(lon), data=fiat, method="REML")

summ.gam<-summary(fit.gam)
summ.gam
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(price) ~ model + owners + power.cat + s(age) + s(km) + s(lat) +
##     s(lon)
##
## Parametric coefficients:
##                 Estimate Std. Error  t value Pr(>|t|)
## (Intercept)     9.034605   0.003179 2842.203  < 2e-16 ***
## modelpop       -0.039718   0.006182   -6.425 1.76e-10 ***
```

```
## modelsport     -0.010112   0.011971   -0.845   0.3984
## owners2         0.005051   0.009581    0.527   0.5981
## owners3         0.008372   0.020831    0.402   0.6878
## owners4        -0.012820   0.033019   -0.388   0.6979
## power.cat60-69  0.005230   0.015574    0.336   0.7370
## power.cat70-79  0.032206   0.017654    1.824   0.0683 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df       F  p-value
## s(age) 3.779  4.685 221.653  < 2e-16 ***
## s(km)  2.924  3.736 136.164  < 2e-16 ***
## s(lat) 4.543  5.569   5.752 1.85e-05 ***
## s(lon) 4.343  5.308   3.550  0.00258 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.856   Deviance explained = 85.8%
## -REML = -1329.9  Scale est. = 0.0096719  n = 1538
```

```
# Putting GAM smooth terms into a nice looking table
pander(summ.gam$s.table,digits=3)
```

	edf	Ref.df	F	p-value
s(age)	3.78	4.68	222	0
s(km)	2.92	3.74	136	0
s(lat)	4.54	5.57	5.75	1.85e-05
s(lon)	4.34	5.31	3.55	0.00258

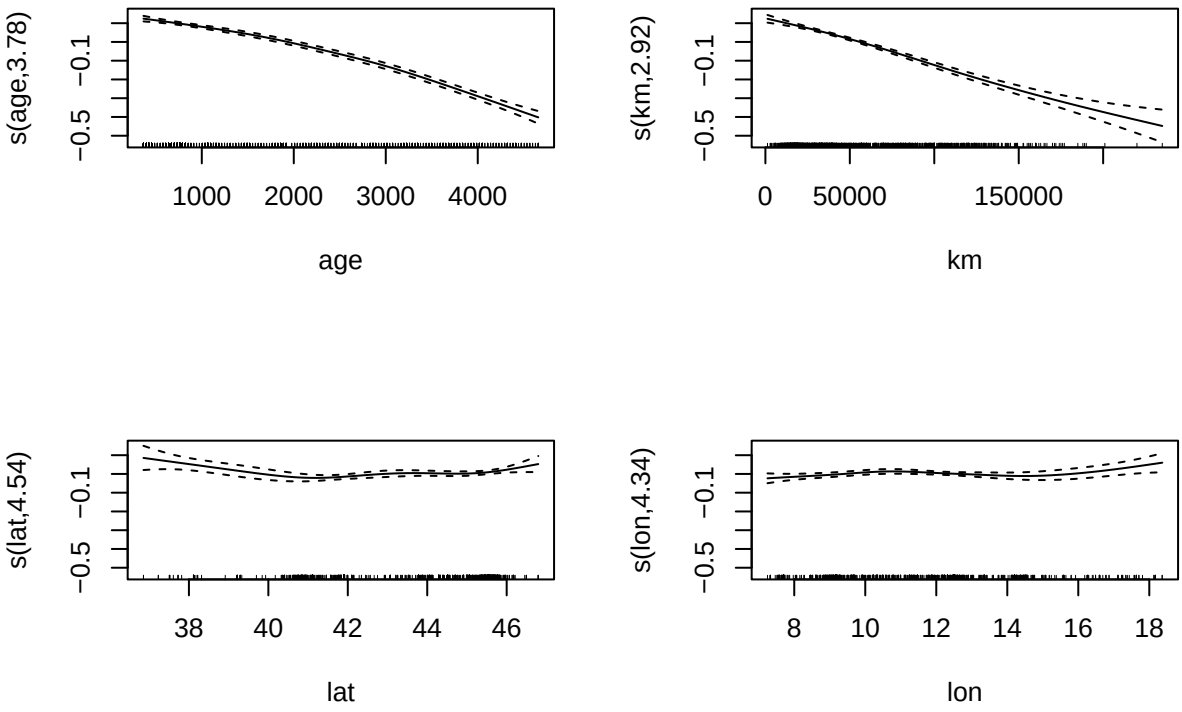
In the model labelled 'fit', there are smooth terms for 4 predictors (`age`, `km`, `lat`, & `lon`). Based on the plots below and the values in the table above, 'age', 'lat' & 'lon' are all non-linear and significant. This is because all smooth terms have a high edf and a low p-value. 'km' has a low edf and a low p-value, meaning that km is linear and significant. Since the p-values for all terms are less than 0.05, we reject the null hypothesis and conclude that the smooth terms for age, km, lat & lon are significant.

Hypotheses

H_0 : the smooth term is not significant

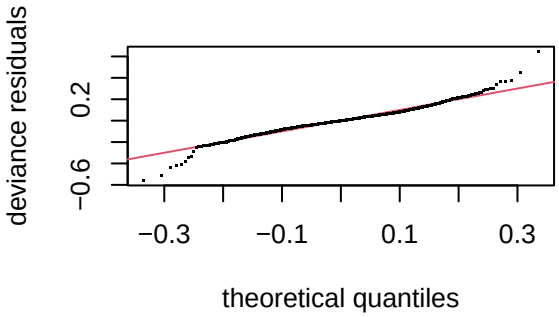
H_1 : the smooth term is significant

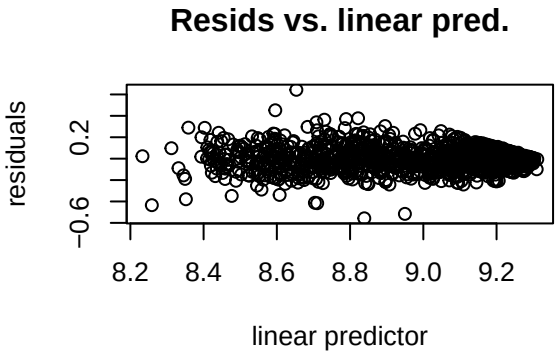
```
par(mfrow=c(2,2))
plot(fit.gam)
```

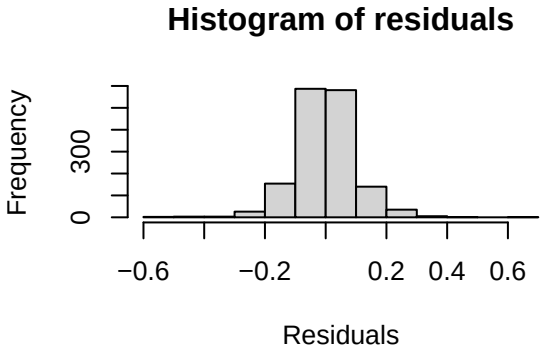


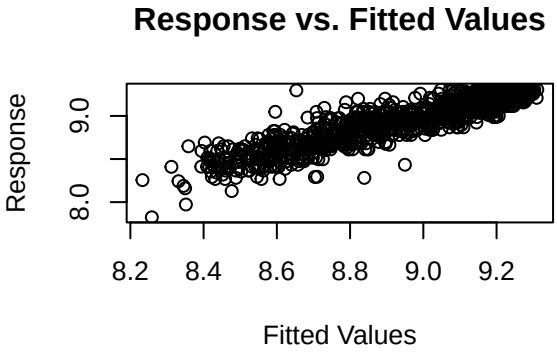
b. **(5 marks)** Perform a diagnostic check of regression assumptions and adequacy of basis functions for the model you fitted in part (a). What conclusions do you draw from your results?

```
# Model checking - residual diagnostics & basis functions
par(mfrow=c(2,2))
gam.check(fit.gam, k.rep=1000)
```










```
##
## Method: REML   Optimizer: outer newton
## full convergence after 6 iterations.
## Gradient range [-8.483807e-06,6.633103e-08]
## (score -1329.857 & scale 0.009671931).
## Hessian positive definite, eigenvalue range [0.4604198,763.0115].
## Model rank =  44 / 44
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k'  edf k-index p-value
## s(age) 9.00 3.78    0.96   0.074 .
## s(km)  9.00 2.92    0.96   0.051 .
## s(lat) 9.00 4.54    0.91  <2e-16 ***
## s(lon) 9.00 4.34    0.92  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Regression assumptions:

Q-Q plot: This plot shows that there seems to be some non-normality present. Deviance residuals heavily deviate from the Q-Q line at the beginning and the end.

Resids vs. linear pred plot: The residuals are relatively equally spread without a distinct pattern. This is good indication that there are no uncaptured signficant non-linear relationships.

Histogram of residuals plot: This plot is showing that residuals are normally distributed.

Response vs. Fitted values plot: This plot shows the overall fit of the model. It looks like constant variance and linearity are present.

Adequacy of basis functions:

The output above reports full convergence, indicating that an optimal solution has been found. The maximum number of basis functions considered (k') is 9. Small p-values indicate that residuals are not randomly distributed. In the model, the p-values for all smooth terms (`age`, `km`, `lat`, & `lon`) are relatively low (lowest for lat and lon), and k-index is close to 1 for all smooth terms. The edf for all smooth terms is a lot lower than k', indicating that we are likely to have adequate numbers of basis functions. Refitting the model with higher k values for `age`, `km`, `lat`, & `lon` could be considered to be sure that there is enough basis functions for all smooth terms.

c. **(4 marks)** For ease of interpretation, a linear model is preferred to a GAM. Fit a linear model (using `lm`) for `log(price)` with predictors as shown in model `fit1`. Based on your fitted model, give a mathematical interpretation of the effect of `age` on `price`.

```
fit1<-lm(log(price) ~ model + power.cat + age + km + owners + lat + lon, data=fiat)
pander(summary(fit1), caption="")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	9.108	0.09992	91.15	0
modelpop	-0.03264	0.006358	-5.133	3.218e-07
modelsport	-0.02236	0.01238	-1.806	0.07109
power.cat60-69	0.02134	0.01582	1.349	0.1776
power.cat70-79	0.002318	0.01755	0.1321	0.8949
age	-0.0001145	3.761e-06	-30.46	1.447e-159
km	-2.449e-06	1.198e-07	-20.44	2.93e-82
owners2	0.00314	0.009968	0.315	0.7528
owners3	0.01189	0.02167	0.5486	0.5834
owners4	-0.01367	0.03442	-0.3972	0.6913
lat	0.0052	0.001919	2.709	0.006819
lon	0.001693	0.001761	0.9617	0.3364

Observations	Residual Std. Error	R^2	Adjusted R^2
1538	0.1027	0.8443	0.8432

Fitted model: $\widehat{log(price)} = 9.108 - 0.037 modelpop - 0.022 modelsport + 0.021 power.cat_{60-69} + 0.002 power.cat_{70-79} - 0.0001145 age - 2.449 * 10^{-}6 km + 0.00314 owners_2 + 0.01189 owners_3 - 0.01367 owners4 + 0.0052 lat + 0.001693 lon$

Mathematical interpretation of the effect of `age` on `price`:

The regression coefficient for `age` is $e^{-}0.0001145 - 1 = -0.00011445$. Therefore, we expect the selling price in euros (€) to decrease by a multiplicative factor of 0.00011445 for each additional day the car ages. When age decreases by 1 while holding all other predictors constant we expect: change in price = price $\times -0.0001145$.

d. **(5 marks)** Use the `step` function to perform stepwise model selection for the model in part (c) based on AIC and BIC, to determine whether any of the predictors can be excluded from the model. List the predictors included in your preferred model in each case and justify your answer.

```
##AIC
step(fit1, direction = "both")
```

```
## Start:  AIC=-6988.83
## log(price) ~ model + power.cat + age + km + owners + lat + lon
##
##             Df Sum of Sq    RSS     AIC
## - owners     3    0.0059 16.101 -6994.3
## - power.cat  2    0.0192 16.115 -6991.0
## - lon        1    0.0098 16.105 -6989.9
## <none>                   16.095 -6988.8
## - lat        1    0.0774 16.173 -6983.4
## - model      2    0.2862 16.382 -6965.7
## - km         1    4.4059 20.501 -6618.7
## - age        1    9.7832 25.879 -6260.5
##
## Step:  AIC=-6994.26
## log(price) ~ model + power.cat + age + km + lat + lon
##
##             Df Sum of Sq    RSS     AIC
## - power.cat  2    0.0191 16.120 -6996.4
## - lon        1    0.0097 16.111 -6995.3
## <none>                   16.101 -6994.3
## - lat        1    0.0767 16.178 -6989.0
## + owners     3    0.0059 16.095 -6988.8
## - model      2    0.2848 16.386 -6971.3
## - km         1    4.4187 20.520 -6623.3
## - age        1    9.7966 25.898 -6265.3
##
## Step:  AIC=-6996.44
## log(price) ~ model + age + km + lat + lon
##
##             Df Sum of Sq    RSS     AIC
## - lon        1    0.0096 16.130 -6997.5
## <none>                   16.120 -6996.4
## + power.cat  2    0.0191 16.101 -6994.3
## - lat        1    0.0770 16.197 -6991.1
## + owners     3    0.0058 16.115 -6991.0
## - model      2    0.2780 16.398 -6974.1
## - km         1    4.4291 20.549 -6625.1
## - age        1    9.9058 26.026 -6261.7
##
## Step:  AIC=-6997.52
## log(price) ~ model + age + km + lat
##
##             Df Sum of Sq    RSS     AIC
## <none>                   16.130 -6997.5
## + lon        1    0.0096 16.120 -6996.4
## + power.cat  2    0.0191 16.111 -6995.3
## + owners     3    0.0058 16.124 -6992.1
## - lat        1    0.0993 16.229 -6990.1
## - model      2    0.2753 16.405 -6975.5
## - km         1    4.4231 20.553 -6626.8
```

```
## - age        1    9.9813 26.111 -6258.7

##
## Call:
## lm(formula = log(price) ~ model + age + km + lat, data = fiat)
##
## Coefficients:
## (Intercept)     modelpop   modelsport          age           km          lat
##   9.189e+00   -3.203e-02   -2.067e-02   -1.143e-04   -2.439e-06    3.779e-03
```

The 'starting' model is the one that includes all predictors and has $AIC = -6997.5$. From the output above we can see that when the predictors lon, power.cat, and owners are included in the model, we get an increase in AIC. When the predictors lat, model, km, and age are removed from the model, we also get an increase in AIC.

$AIC(A) - AIC(B) = -6996.4 - (-6997.5) = 1.1$

The difference is in the [0,2.5) interval. Therefore, applying the AIC rules of thumb means there is no difference in models, and we prefer the model with fewer predictors using the principle of parsimony.

The preferred model according to the AIC criterion includes the four predictors: `model, age, km` and `lat`.

```
##BIC
step(fit1, direction="both", k=log(nrow(fiat)))
```

```
## Start:  AIC=-6924.77
## log(price) ~ model + power.cat + age + km + owners + lat + lon
##
##             Df Sum of Sq    RSS     AIC
## - owners     3    0.0059 16.101 -6946.2
## - power.cat  2    0.0192 16.115 -6937.6
## - lon        1    0.0098 16.105 -6931.2
## <none>                   16.095 -6924.8
## - lat        1    0.0774 16.173 -6924.7
## - model      2    0.2862 16.382 -6912.3
## - km         1    4.4059 20.501 -6560.0
## - age        1    9.7832 25.879 -6201.7
##
## Step:  AIC=-6946.22
## log(price) ~ model + power.cat + age + km + lat + lon
##
##             Df Sum of Sq    RSS     AIC
## - power.cat  2    0.0191 16.120 -6959.1
## - lon        1    0.0097 16.111 -6952.6
## - lat        1    0.0767 16.178 -6946.3
## <none>                   16.101 -6946.2
## - model      2    0.2848 16.386 -6933.9
## + owners     3    0.0059 16.095 -6924.8
## - km         1    4.4187 20.520 -6580.6
## - age        1    9.7966 25.898 -6222.6
##
## Step:  AIC=-6959.07
## log(price) ~ model + age + km + lat + lon
##
##             Df Sum of Sq    RSS     AIC
```

```
## - lon        1    0.0096 16.130 -6965.5
## - lat        1    0.0770 16.197 -6959.1
## <none>                   16.120 -6959.1
## - model      2    0.2780 16.398 -6947.4
## + power.cat  2    0.0191 16.101 -6946.2
## + owners     3    0.0058 16.115 -6937.6
## - km         1    4.4291 20.549 -6593.1
## - age        1    9.9058 26.026 -6229.7
##
## Step:  AIC=-6965.49
## log(price) ~ model + age + km + lat
##
##             Df Sum of Sq    RSS     AIC
## <none>                   16.130 -6965.5
## - lat        1    0.0993 16.229 -6963.4
## + lon        1    0.0096 16.120 -6959.1
## - model      2    0.2753 16.405 -6954.1
## + power.cat  2    0.0191 16.111 -6952.6
## + owners     3    0.0058 16.124 -6944.0
## - km         1    4.4231 20.553 -6600.1
## - age        1    9.9813 26.111 -6232.0


##
## Call:
## lm(formula = log(price) ~ model + age + km + lat, data = fiat)
##
## Coefficients:
## (Intercept)     modelpop   modelsport          age           km          lat
##   9.189e+00   -3.203e-02   -2.067e-02   -1.143e-04   -2.439e-06    3.779e-03
```

The model that includes all predictors is to be labelled model B, and the model with the next largest BIC value (excludes `lat`) is labelled model A.

$BIC(A) - BIC(B) = -6963.4 - (-6965.5) = 2.1$

As the difference is within the interval of [2.0,6.0), using Raftery BIC rules of thumb means there is positive preference for model B, the model with a smaller BIC value. Therefore, we opt for the model with more predictors, model B. Therefore, the preferred model according to the BIC criterion includes all predictors: `model`, `power.cat`, `age`, `km`, `owners`, `lat`, and `lon`.

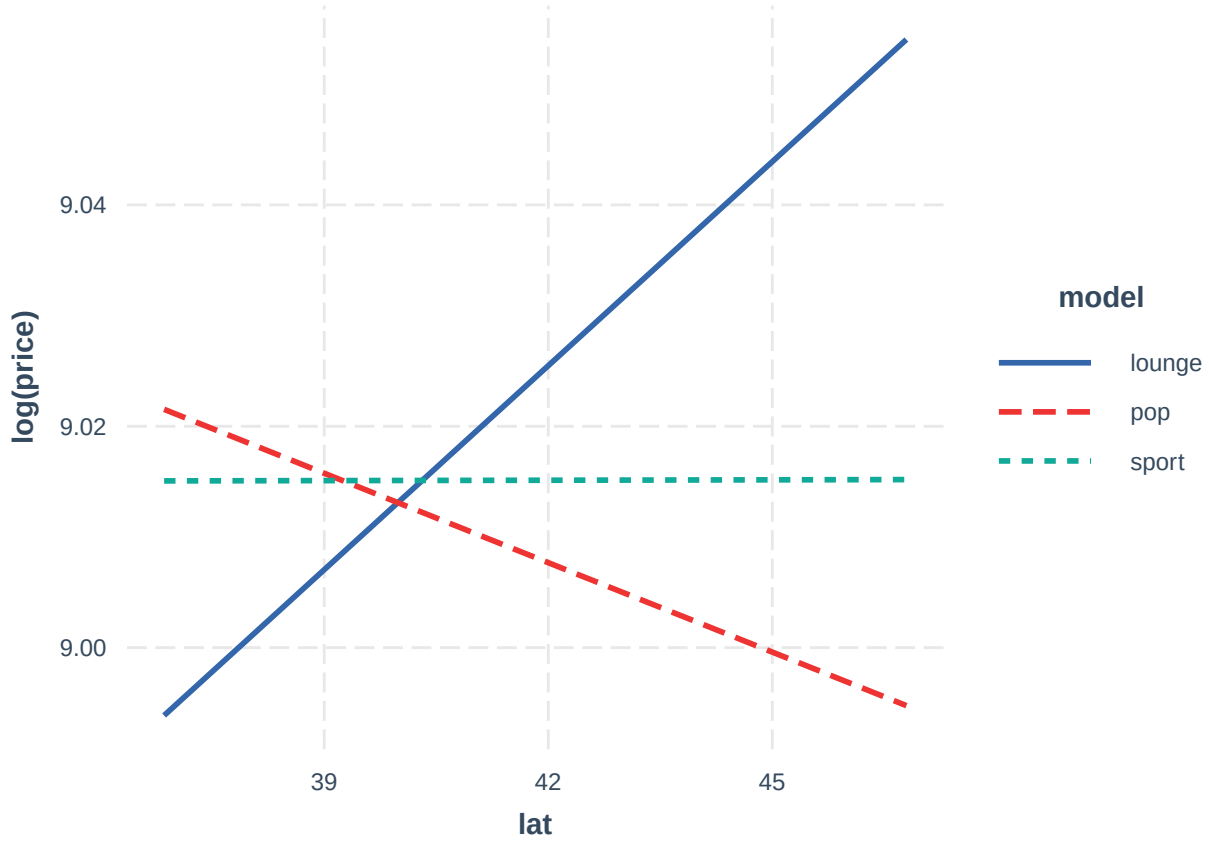
e. **(4 marks)** It is known that the price of cars in Italy varies from North to South of the country. You also suspect that the effect of `lat` varies by `model`, and you therefore investigate the interaction between `lat` and `model`. Add the interaction `model:lat` to the preferred model based on *AIC* in part (d). Obtain an interaction plot and use it to describe briefly the effect of `lat` on `log(price)`

```
library(interactions)
```

```
## Adding an interaction term (model:lat) to the preferred model based on AIC
fit2<-lm(log(price) ~ model + age + km + lat + model:lat, data=fiat)
pander(summary(fit2)$coefficients, caption="")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	9.086	0.06314	143.9	0
modelpop	0.3533	0.1284	2.751	0.006012
modelsport	0.2472	0.2378	1.039	0.2988
age	-0.0001146	3.708e-06	-30.9	2.726e-163
km	-2.432e-06	1.188e-07	-20.46	1.891e-82
lat	0.006144	0.001451	4.235	2.422e-05
modelpop:lat	-0.008837	0.002941	-3.004	0.002706
modelsport:lat	-0.006132	0.005424	-1.13	0.2585

```
## Interaction plot
interact_plot(fit2,pred=lat, modx=model,
              colors = "Qual1", data=fiat)
```




```
addmargins(table(fiat$model))
```

```
##
## lounge    pop  sport    Sum
##   1094    358     86   1538
```

The interaction plot above shows that the effect of `lat` varies by model.

Null hypothesis $H_0 : lat = 0$ (there is no interaction)

Alternative hypothesis $H_1 : lat \neq 0$ (there is an interaction)

For there interaction between modelpop and lat, the p-value of $0.0027 < 0.05$, thus there is evidence to reject the null hypothesis and conclude there is an interaction between modelpop and lat. For the interaction between modelsport and lat, the p-value of $0.2585 > 0.05$, thus there is not enough evidence to reject the null hypothesis, therefore there is not enough evidence to conclude an interaction between modelsport and lat.

The plot shows that for model(lounge), an increase in lat is associated with an increase in (log)price. For model(pop), an increase in lat is associated with a reduction in log(price). For model(sport), an increase in lat is associated with no change in log(price). This might be due to there being 86 observations in the dataset for modelsport. Modelsport has the least amount of observations, as modellounge has 1094 and modelpop has 358 observations.

f. **(4 marks)** Obtain and print in a table, the *AIC* and *BIC* values for your *AIC*-based preferred model in part (d) and the model in part (e). Based on these values, state whether or not you would include the interaction term in your final preferred model and justify your answer.

```
# The model fit2 has the interaction term
# The model fit3 does not have the interaction term

## Fitting the model from part d (the AIC-based preferred model)
fit3<-lm(log(price) ~ model + age + km + lat, data=fiat)

##AIC
pander(AIC(fit2, fit3), caption="")
```

	df	AIC
fit2	9	-2636
fit3	7	-2631

```
##BIC
pander(BIC(fit2, fit3), caption="")
```

	df	BIC
fit2	9	-2588
fit3	7	-2593

Based off of the AIC and BIC values, results show that the model with the interaction term (fit 2) (lower AIC) is the preferred model according to AIC, while the model without the interaction term (fit3) (lower BIC) is the preferred model according to BIC.

Since we are aiming to find the key predictors of price and in understanding how these how these predictors affect price in the fiat dataset, this is an inference problem. We can use adjusted R^2 to find out which model is preferred.

```
fit2.adjrsq<-summary(fit2)$r.sq
fit3.adjrsq<-summary(fit3)$r.sq

modname<-c("fit2", "fit3")
adj.rsq<-c(fit2.adjrsq,fit3.adjrsq)
rsq<-data.frame(modname,adj.rsq)
pander(rsq, caption="")
```

modname	adj.rsq
fit2	0.845
fit3	0.844

We see that the model with the interaction term (fit2) has a slightly higher adjusted R^2 compared to the model without the interaction term (fit3). We opt for the model with the interaction term as the preferred model.

Q2. (14 marks) In this next question we'll focus on constructing a prediction model for the Fiat data using subset selection and shrinkage methods.

a. **[6 marks]** Use the `olsrr` package to perform best subset, forward and backward stepwise model selection for the Fiat data. In each case, use AIC as the model performance metric to base your selection on and list the predictors in your final model.

```
library(olsrr)
```

```
# Using the fiat data, I fit a linear model that includes all predictors
fit.full<-lm(price ~ ., data = fiat)
```

```
# Best subset
best.subset<-ols_step_best_subset(fit.full)
# We now have the best models M1,...M7 with 1, 2, 3,...,7 predictors
sub.aic<-best.subset$aic # Select the best model out of the models shown in best.subset using AIC metri
which.min(sub.aic) # This gives the model with the lowest AIC
```

```
## [1] 4
```

```
pander(best.subset$predictors[4])
```

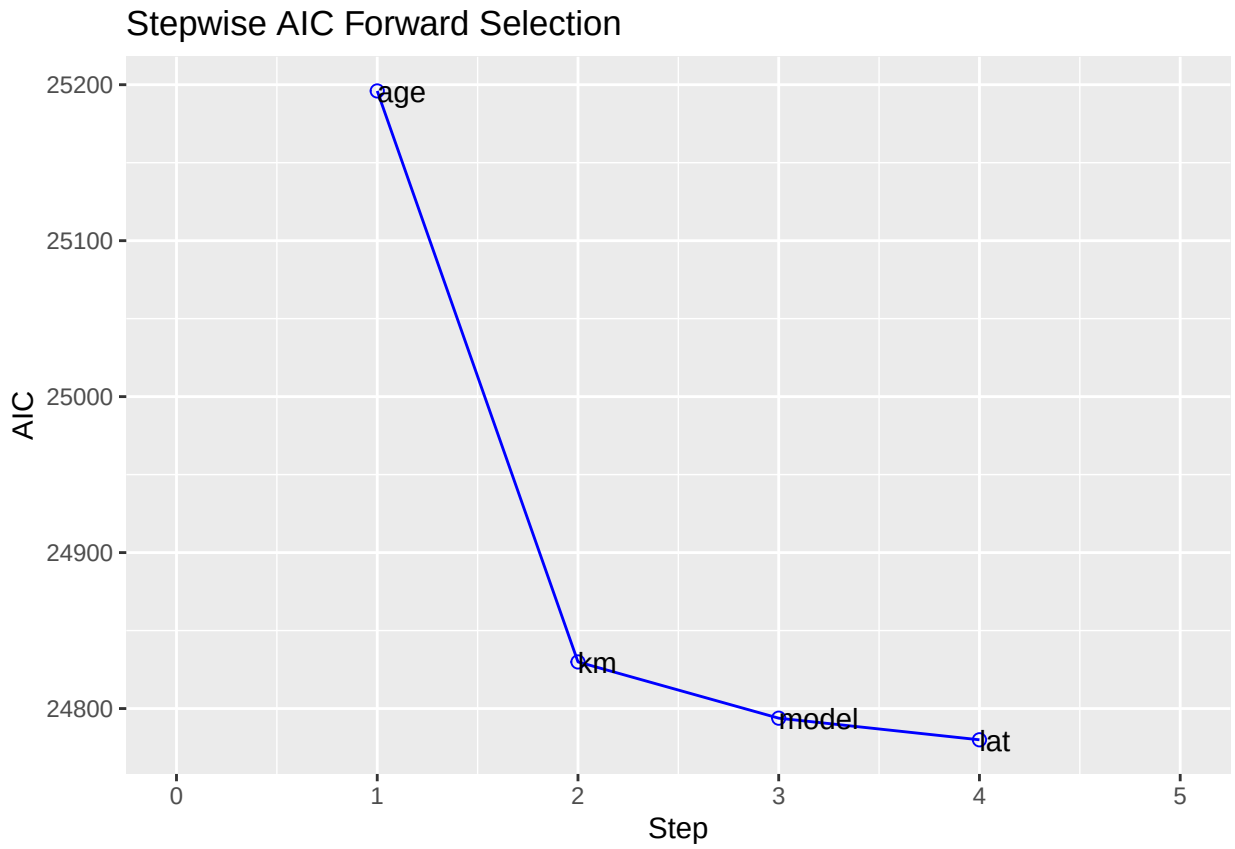
model age km lat

The model with the lowest AIC and therefore the best model selected using best subset includes 4 predictors (model, age, km, and lat) (M4). The AIC for this model (M4) is 24779.96, yet the model (M5) with the second lowest AIC includes model, power.cat, age, km, and lat. M5 obtained an AIC of 24780.51, which is not much higher than the model with the lowest AIC. This is due to the AIC's penalty for model complexity 2(p+1). While adding more predictors (in this case model.cat) can sometimes improve model fit, it can also lead to overfitting or increased complexity, so we prefer M4 with fewer predictors.

```
# Forward selection
forward.stepwise<-ols_step_forward_aic(fit.full)
forward.stepwise
```

```
##
##                                Selection Summary
## ------------------------------------------------------------------------------------
## Variable        AIC           Sum Sq              RSS           R-Sq      Adj. R-Sq
## ------------------------------------------------------------------------------------
## age          25196.001    4616153663.031    1168252892.945    0.79803      0.79790
## km           24829.893    4864821538.691     919585017.285    0.84102      0.84082
## model        24793.825    4888468845.057     895937710.920    0.84511      0.84471
## lat          24779.961    4897662696.300     886743859.676    0.84670      0.84620
## ------------------------------------------------------------------------------------
```

```
plot(forward.stepwise)
```




```
ols_step_forward_aic(fit.full)$model
```

```
##
## Call:
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
##     data = l)
##
## Coefficients:
## (Intercept)          age           km     modelpop   modelsport          lat
##  9459.31655     -0.87205     -0.01786   -303.23663    -43.22406     36.35622
```

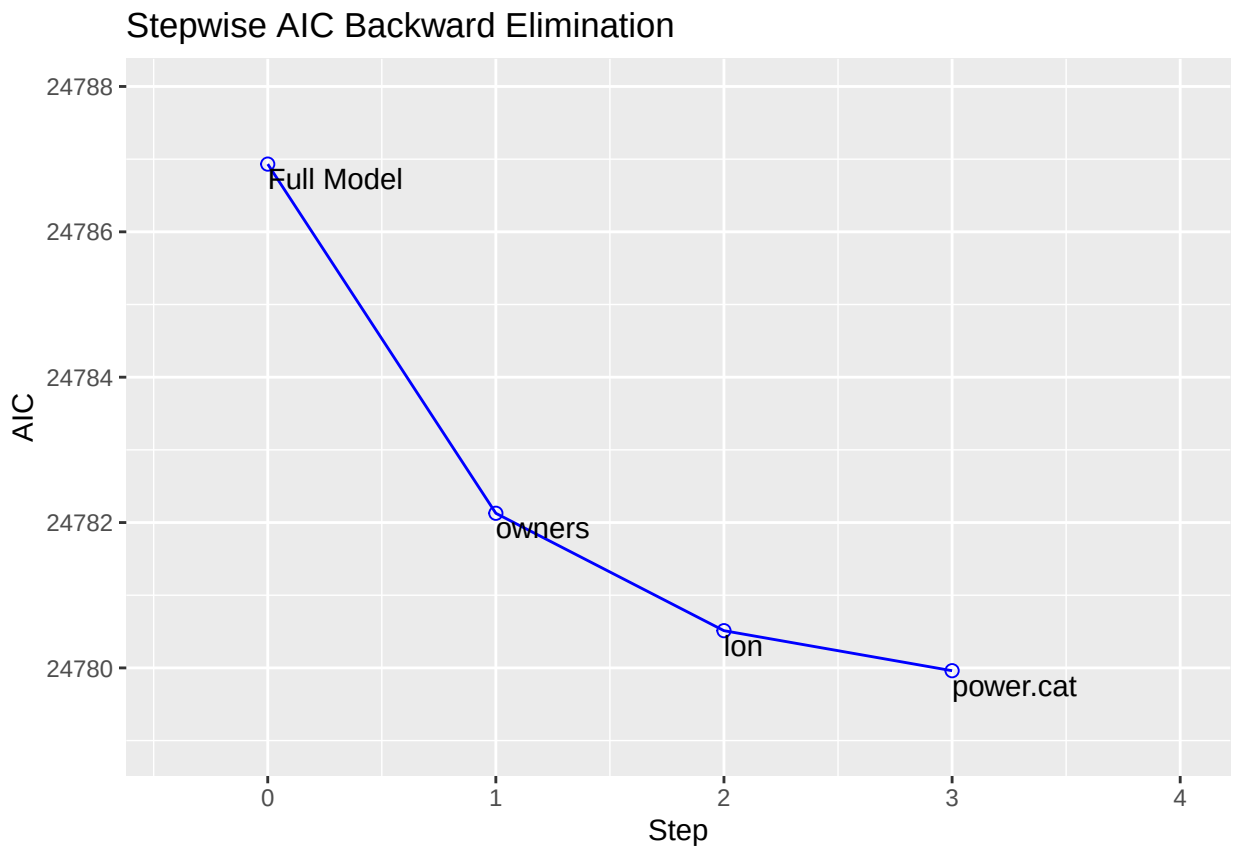
The best model for forward stepwise model selection includes 4 predictors being age, km, model, and lat, where the predictors are added to the model in this order. The plot shows the variables added to the model at each step (starting from a null model with no predictors).

```
# Backward selection
backward.stepwise<-ols_step_backward_aic(fit.full)
backward.stepwise
```

```
##
##
##                            Backward Elimination Summary
```

```
## ---------------------------------------------------------------------------------------
## Variable         AIC            RSS             Sum Sq         R-Sq      Adj. R-Sq
## ---------------------------------------------------------------------------------------
## Full Model    24786.932    883848901.476    4900557654.501    0.84720      0.84610
## owners        24782.128    884536285.852    4899870270.124    0.84708      0.84628
## lon           24780.512    884757386.596    4899649169.381    0.84704      0.84634
## power.cat     24779.961    886743859.676    4897662696.300    0.84670      0.84620
## ---------------------------------------------------------------------------------------
```

```
plot(backward.stepwise)
```




```
ols_step_backward_aic(fit.full)$model
```

```
##
## Call:
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
##     data = l)
##
## Coefficients:
## (Intercept)     modelpop   modelsport          age           km          lat
##  9459.31655   -303.23663    -43.22406     -0.87205     -0.01786     36.35622
```

The best model for backward stepwise model selection includes 4 predictors being model, age, km, and lat (backward selection found this by excluding owners, lon, and power.cat when starting from a full model that included all predictors). The predictors were excluded from the model in the order: owners, lon, power.cat.

The final model includes 4 predictors being `model`, `age`, `km`, and `lat`. This is because when using AIC as the performance metric, best subset selection, forward selection, and backward selection all chose the same model.

b. **[3 marks]** Apply ridge regression to the Fiat data and use cross-validation to identify the "best" value, λ_{MSEmin}, for the penalty parameter λ. In a table, print the coefficients for a model fitted using λ_{MSEmin} and a model fitted using $\lambda = 0$. Based on your table are there any predictors that you would consider for exclusion? Explain your answer briefly.

```
library(glmnet)
```

```
## Create the design matrix
x<-model.matrix(price~.,fiat)
x[1:3,]
```

```
##   (Intercept) modelpop modelsport  age     km owners2 owners3 owners4      lat
## 1           1        0          0  882  25000       0       0       0 44.90724
## 2           1        1          0 1186  32500       0       0       0 45.66636
## 3           1        0          1 4658 142228       0       0       0 45.50330
##        lon power.cat60-69 power.cat70-79
## 1  8.61156              0              0
## 2 12.24189              0              0
## 3 11.41784              0              1
```

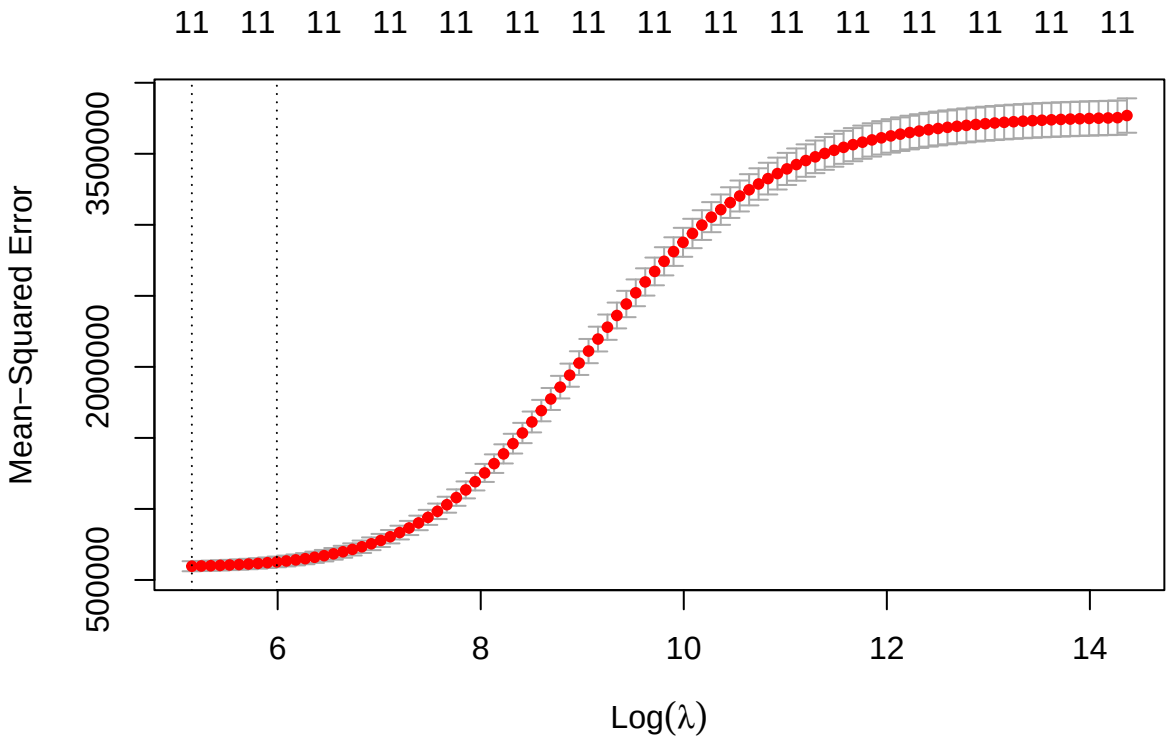
```
## Exclude the column of 1's that represent the intercept
x<-x[, -1]
x[1:3,]
```

```
##   modelpop modelsport  age     km owners2 owners3 owners4      lat      lon
## 1        0          0  882  25000       0       0       0 44.90724  8.61156
## 2        1          0 1186  32500       0       0       0 45.66636 12.24189
## 3        0          1 4658 142228       0       0       0 45.50330 11.41784
##   power.cat60-69 power.cat70-79
## 1              0              0
## 2              0              0
## 3              0              1
```

```
## Create the y vector
y<-fiat$price

## Fit the ridge regression model
ridge.mod<-glmnet(x,y,alpha=0) # alpha=0 for ridge regression

## Using CV to identify the "best" value
set.seed(1)
cv_ridge <- cv.glmnet(x = x,y = y, alpha = 0)
plot(cv_ridge)
```




```
cv_ridge$lambda.min ## best value of the penalty parameter lambda
```

```
## [1] 173.2455
```

```
## Fit model using "best" lambda
ridge.mod.best <-glmnet(x,y, alpha = 0,lambda = cv_ridge$lambda.min)

## Fit model using lambda=0
ridge.mod.zero <-glmnet(x,y, alpha = 0,lambda = 0)

## Produce table of coefficients of both models
pander(data.frame("Best" = coef(ridge.mod.best)[, 1],
"Lambda=0" = coef(ridge.mod.zero)[, 1]),
col.names = c("Best", "Lambda=0"))
```

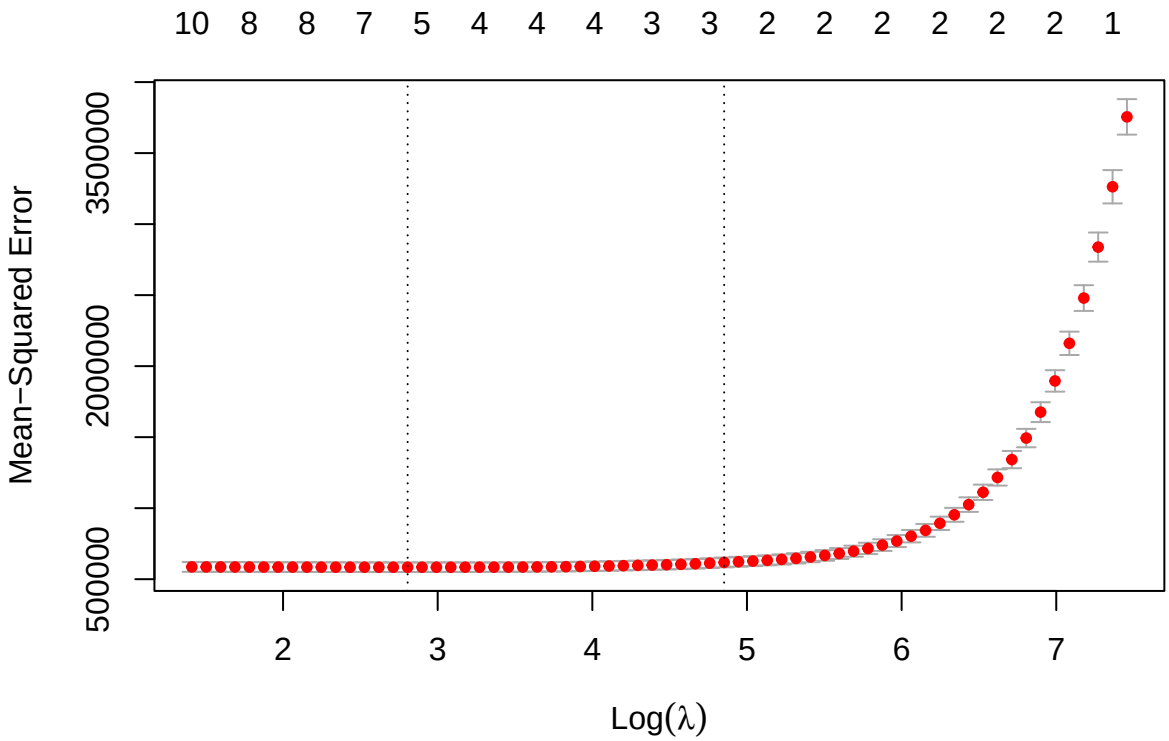
	Best	Lambda=0
(Intercept)	9457	9050
modelpop	-306.7	-302.9
modelsport	-163.1	-65.7
age	-0.7797	-0.8782
km	-0.01852	-0.01797
owners2	27.99	64.28
owners3	39.6	42.79

	Best	Lambda=0
owners4	-153	-149.5
lat	33.16	43.72
lon	2.596	8.138
power.cat60-69	-66.49	16.97
power.cat70-79	73.95	241.8

All estimated coefficients are non-zero, indicating that all predictors are required for the model, regardless of which values for λ was used. Coefficents cannot be completely shrunk to zero with ridge regression unless $\lambda = infinity$. A sensible choice is $\lambda_{MSEmin} = 173.2455$. Shrinkage has not had the same impact on all regression coefficients. For example for modelsport, as lambda decreased to zero, shrinkage toward zero happens, whereas for lat, as lambda was decreased to zero, the coefficient increased.

c. **[3 marks]** Apply lasso regression to the Fiat data and use cross-validation to identify the "best" value, λ_{MSEmin}, for the penalty parameter λ. In a table, print the coefficients for a model fitted using λ_{MSEmin} and a model fitted using $\lambda = 0$. Based on your table identify predictors that you would consider for exclusion. Explain your answer briefly.

```
## Fit lasso regression model
lasso.mod <- glmnet(x, y, alpha = 1)
set.seed(1)
cv_lasso <- cv.glmnet(x = x,y = y, alpha = 1)
plot(cv_lasso)
```



```
cv_lasso$lambda.min
```

```
## [1] 16.53712
```

```
## Fit model using "best" lambda
lasso.mod.best <-glmnet(x,y, alpha = 1,lambda = cv_lasso$lambda.min)

## Fit model using lambda=0
lasso.mod.zero <-glmnet(x,y, alpha = 1,lambda = 0)

## Produce a table
pander(data.frame("Best" = coef(lasso.mod.best)[, 1],
"Lambda=0" = coef(lasso.mod.zero)[, 1]),
col.names = c("Best", "Lambda=0"))
```

	Best	Lambda=0
(Intercept)	9794	9050
modelpop	-262.1	-302.9
modelsport	0	-65.7
age	-0.8686	-0.8782
km	-0.01773	-0.01797
owners2	0	64.28
owners3	0	42.79
owners4	0	-149.5
lat	28.05	43.72
lon	0	8.138
power.cat60-69	0	16.97
power.cat70-79	82.79	241.8

When the model is using λ_{MSEmin}, coefficients are shrunk to zero for the predictors modelsport, owners2, owners3, owners4, lon, and power.cat60-69. This means that when using the 'best' λ, these predictors can be excluded from the model. When $\lambda = 0$, no predictors can be excluded from the model, as no coefficients are completely shrunk to zero.

d. **[2 marks]** Based on your results in parts (b) and (c), which of the two approaches, ridge regression or lasso regression, would you prefer to use for model selection. Explain your answer briefly.

I would prefer to use lasso regression over ridge regression for model selection because the aim is to select the best predictors. Ridge regression is not used to select predictors, but is a shrinkage method that shrinks coefficients. Ridge regression does not create a sparser model, and has no clear way of indicating which predictors to be excluded. Therefore, lasso regression is better to use for model selection as it is more likely to be able to find the best predictive model as coefficients can be completely shrunk to zero, thus sparse models can be created. Additionally, lasso regression somewhat aligns with the results obtained with backward selection, where lon can be excluded from the model, and levels of owners and power.cat can be excluded. With ridge regression, predictors have not been excluded like lasso regression and backward selection have.

Assignment total: 40 marks