Bayesian Assignment 2

Gavin Connolly

25/02/2022

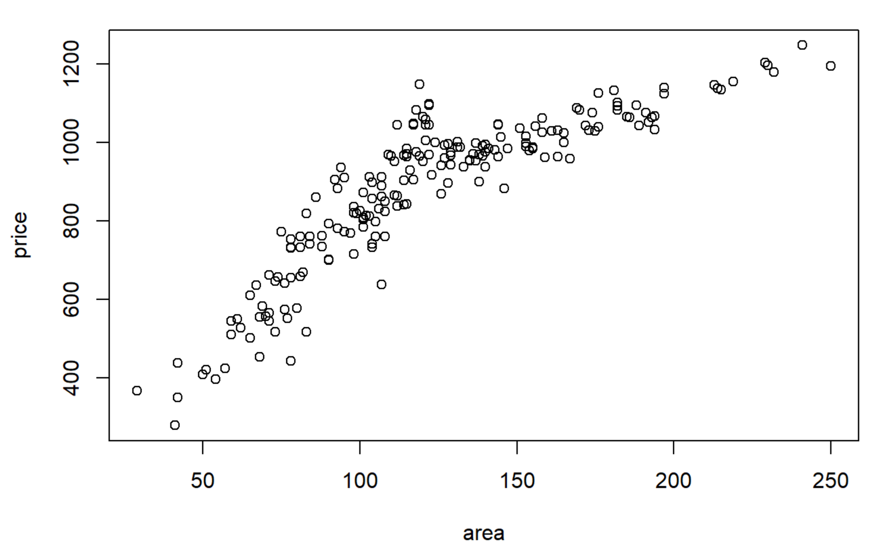
Introduction

The dataset which you will analyse in this assignment involves data of 200 house prices (in units of 1, 000 euro) in a particular suburb of Co. Dublin. The floor area of each house (in square metres) is also given

```
dublinproperty <- read.csv("property.csv")

str(dublinproperty) # Look at summary of data
```

```
## 'data.frame':    200 obs. of  3 variables:
##  $ X    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ area : int  117 112 153 138 104 140 29 143 165 176 ...
##  $ price: num  1046 864 998 968 732 ...
```

```
area <- dublinproperty$area
price <- dublinproperty$price
plot(area, price) # look at plot of the data to check for patterns
```




```
# looks as if a regression line through data would cross y-axis at price of roughly 300
# may be a good centre for our prior on the alpha/intercept term of our model
```

The objective here is to develop a Bayesian linear regression model and implement it in Stan.

- Describe clearly your model including the priors that you have chosen. Generally, you should choose priors that are quite vague and uninformative.
- Produce a Stan program to fit this model. Present and interpret the results from this model.
- Plot the data together with 90% posterior credible intervals for the posterior mean line as well as 90% credible interval for the posterior predicted prices.
- Are you happy with the model fit? If not, explain why and suggest changes that could be made to the model to rectify this.

For my priors, I have chosen the following for each of the parameters in the model:

- α, the intercept term for our linear regression model for the mean μ: Uniform prior, taking values on the interval 0 to 500.

  We have no real information on what the prior for the intercept should be other than it should be greater than 0. I have then chosen an upper bound of 500 as €500,000 seems like a decent upper bound for the minimum price for a house (or in this case a house with 0 area).

- β, the slope term for our linear regression model for the mean μ: Normal prior, w/ mean 0 & standard deviation of 10.

  I have chosen this prior as it is centred around 0, which assumes there is no linear relationship between the 2 variables; area & price. I have then chosen a standard deviation of 10 as it allows the parameter to take a relatively wide range of values & provide flexibility for the posterior estimates.

- σ, the standard deviation for our model of the price: Uniform prior on the interval (0, 200).

  This prior assumes σ takes values between 0 & 200 with equal probability. This is a relatively non-informative prior to account for the uncertainty in the parameter. We have then calculated a value μ, the mean of the distribution of house prices for a given area using the regression equation: $\mu_i = \alpha + \beta \cdot x_i$

The house price is then computed using a normal distribution with parameters μ_i & σ.

```
d <- dublinproperty

dat <- list(N = NROW(dublinproperty), area = area, price = price)

writeLines(readLines("LinRegModel.stan"))
```

```
##
## data {
##   int<lower=1> N;
##   vector[N] price;
##   vector[N] area;
## }
## parameters {
##   real alpha;
##   real beta;
##   real<lower=0,upper=200> sigma; // uniform prior on sigma
## }
## model {
##   vector[N] mu = alpha + beta * area;  //
##   target += normal_lpdf(price | mu, sigma);
##   target += uniform_lpdf(alpha | 0, 500); // prior centred around 0 with high standard deviation
##   target += normal_lpdf(beta | 0, 10); // 0 centred prior assumes no relation
## }
```

```
fit_1 <- stan(file="LinRegModel.stan", data=dat, iter=5000)

monitor(fit_1) # rhat = 1 for each variable, chain converged
```

```
## Inference for the input samples (4 chains: each with iter = 5000; warmup = 0):
##
##             Q5     Q50     Q95    Mean   SD  Rhat Bulk_ESS Tail_ESS
## alpha   357.5   391.4   426.3   391.9 21.1     1     3991     4608
## beta      3.7     4.0     4.2     4.0  0.2     1     3910     4322
## sigma    92.1    99.8   108.7   100.1  5.0     1     5231     4885
## lp__  -1212.2 -1209.6 -1208.5 -1209.8  1.2     1     3596     4902
##
## For each parameter, Bulk_ESS and Tail_ESS are crude measures of
## effective sample size for bulk and tail quantities respectively (an ESS > 100
## per chain is considered good), and Rhat is the potential scale reduction
## factor on rank normalized split chains (at convergence, Rhat <= 1.05).
```

From examining our model results, we see that we have 90% credible interval for the posterior distribution of the α parameter of between 356.7 & 425.8. This translates to the distribution of the mean house price for properties with 0 area, with our estimate of the mean price of such a property being between €356,700 & €425,800, with 90% probability. We see that our 90% credible interval for the posterior distribution of our β parameter shows that β lies between 3.7 & 4.3 with a probability of 90%. This indicates that for each extra square metre in the area of a house, the mean price of the property increases by between €3,700 & €4,300. The posterior credible interval for the σ parameter indicates that the standard deviation of our normally distributed pricing model lies between 92.1 & 109.2 with 90% probability.

```
post <- as.data.frame(fit_1)
cor(cbind(post$alpha,post$beta)) # Correlation between alpha and beta reveals strong correlation
```

We see from the plot that the model does not seem to fit the data particularly well. We see a high number of datapoints which are far outside of the 90% credible interval for the posterior predicted prices, with a particularly large cluster of such datapoints around the 120 m2 point. Thus, I am not happy with the fit of this model as it does not align with the data very well. In order to rectify this, we could consider rescaling the data using some transformation function & refit the model using these rescaled values. Alternatively, we could use some more complex model for the data, perhaps taking quadratic terms into account to account for the apparent non-linear relationship between the variables.

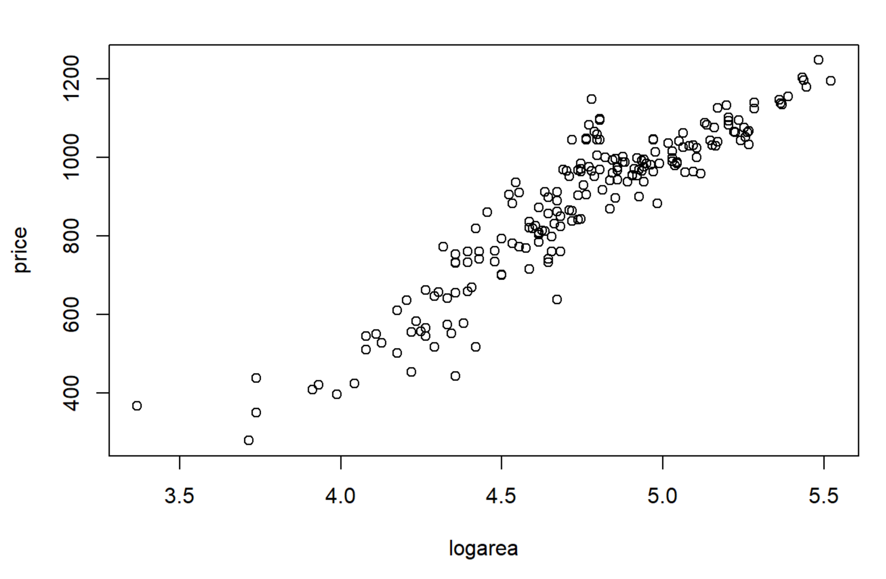
```
f_mu <- function(x) post$alpha + post$beta * x
area_new <- seq(0, 300) # plot showed data lying on interval (0,300)
mu1 <- sapply(area_new, f_mu)

# calculate 90% credible interval for regression mean
y_hdi = HDInterval::hdi(mu1, credMass=0.9)
hpdi_l = y_hdi[1,]
hpdi_u = y_hdi[2,]

p <- ggplot()
# store plot of regression line as well as 90% credible interval for regression mean
p2 <- p +
  geom_point(data = d,
             aes(area, price), shape = 1, color = 'dodgerblue') +
  geom_ribbon(aes(area_new, ymin = hpdi_l, ymax = hpdi_u),
              alpha = .1) +
  geom_abline(data = post,
              aes(intercept = mean(alpha), slope = mean(beta))) +
  labs(subtitle="HPDI Interval = 0.9")

y_pi <- sapply(area_new,
               function(x) rnorm(NROW(post), post$alpha + post$beta * x, post$sigma)
)

# calculate 90% credible intervals for prediction intervals
y_phdi = HDInterval::hdi(y_pi, credMass=0.9)
pi_l = y_phdi[1,]
pi_u = y_phdi[2,]

# plot prediction intervals along with regression line & credible interval for linear mean
p2 + geom_ribbon(mapping = aes(area_new, ymin=pi_l, ymax=pi_u), alpha = 0.05) +
  labs(subtitle = 'Prediction Intervals = 0.9')
# plot shows data far outside of 90% prediction interval at beginning/middle of data range
# data does not follow regression line very closely
```

Question 2

Let's now modify the posterior model by instead modelling the relationship between house price and log(area).

- Implement this model using Stan. Interpret the output from this model.
- Similar to Q1, provide a 90% credible interval for the mean and also a 90% posterior prediction intervals. Plot data overlaying the posterior mean credible interval and also the posterior predictive interval.
- Provide a brief commentary on your analysis.

```
logarea <- log(area)
d2 <- as.data.frame(matrix(data = c(logarea, price), nrow = 200, ncol = 2, byrow = FALSE))
dat2 <- list(N = NROW(dublinproperty), logarea = logarea, price = dublinproperty$price)

plot(logarea, price)
```




```
# harder to see where regression line would cross y-axis in this case
# prior with high variance centred around -1000 seems appropriate
```

```
writeLines(readLines("LogLinRegModel.stan"))
```

```
##
## data {
##   int<lower=1> N;
##   vector[N] price;
##   vector[N] logarea;
## }
## parameters {
##   real alpha;
##   real beta;
##   real<lower=0,upper=200> sigma;
## }
## model {
##   vector[N] mu = alpha + beta * logarea;
##   target += normal_lpdf(price | mu, sigma);
##   target += normal_lpdf(alpha | 0, 500); // value of 1 corresponds to price of 0
##   target += normal_lpdf(beta | 0, 200); // assume no relation (centred on 0)
## }
```

```
fit_2 <- stan(file="LogLinRegModel.stan", data=dat2, iter=5000)

monitor(fit_2) #rhat = 1 for all variables, meaning chain has converged
```

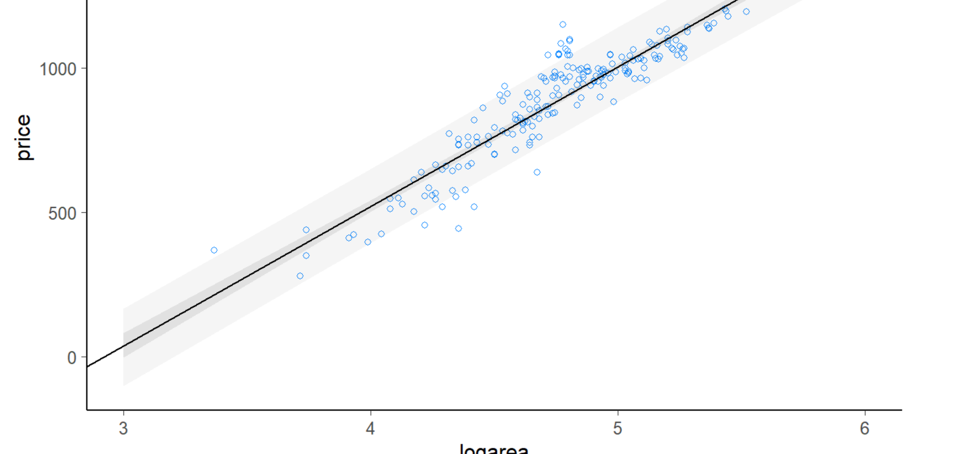
```
## Inference for the input samples (4 chains: each with iter = 5000; warmup = 0):
##
##             Q5     Q50     Q95    Mean   SD  Rhat Bulk_ESS Tail_ESS
## alpha -1524.3 -1408.5 -1295.5 -1409.3 69.2     1     2927     3385
## beta    458.8   482.7   506.9   482.8 14.6     1     2923     3372
## sigma    71.7    77.7    84.7    77.9  4.0     1     3454     3606
## lp__  -1172.9 -1170.2 -1169.2 -1170.5  1.3     1     3028     3835
##
## For each parameter, Bulk_ESS and Tail_ESS are crude measures of
## effective sample size for bulk and tail quantities respectively (an ESS > 100
## per chain is considered good), and Rhat is the potential scale reduction
## factor on rank normalized split chains (at convergence, Rhat <= 1.05).
```

```
post <- as.data.frame(fit_2)
cor(cbind(post$alpha,post$beta)) # Correlation between alpha and beta reveals strong correlation between each.
```

```
##           [,1]      [,2]
## [1,]  1.000000 -0.996874
## [2,] -0.996874  1.000000
```

It appears from the graph that this new linear model fits the data much better than our initial model, but we can also see that there remains a cluster of outlying datapoints around the 120 m^2 mark. Overall, the model seems to be a decent fit for the data, with the majority of the datapoints lying on/around the regression line.

```
f_mu <- function(x) post$alpha + post$beta * x
area_new <- seq(3, 6) # plot seemed to have data lying on interval (3,6)
mu1 <- sapply(area_new, f_mu)

# calculate 90% credible interval for regression mean
y_hdi = HDInterval::hdi(mu1, credMass=0.9)
hpdi_l = y_hdi[1,]
hpdi_u = y_hdi[2,]

p <- ggplot()
# store plot of regression line as well as 90% credible interval for regression mean
p2 <- p +
  geom_point(data = d2,
             aes(logarea, price), shape = 1, color = 'dodgerblue') +
  geom_ribbon(aes(area_new, ymin = hpdi_l, ymax = hpdi_u),
              alpha = .1) +
  geom_abline(data = post,
              aes(intercept = mean(alpha), slope = mean(beta))) +
  labs(subtitle="HPDI Interval = 0.9")

y_pi <- sapply(area_new,
               function(x) rnorm(NROW(post), post$alpha + post$beta * x, post$sigma)
)

# calculate 90% credible intervals for prediction intervals
y_phdi = HDInterval::hdi(y_pi, credMass=0.9)
pi_l = y_phdi[1,]
pi_u = y_phdi[2,]

# plot prediction intervals along with regression line & credible interval for linear mean
p2 + geom_ribbon(mapping = aes(area_new, ymin=pi_l, ymax=pi_u), alpha = 0.05) +
  labs(subtitle = 'Prediction Intervals = 0.9')
```



Question 3

Following from Q2, use the Bayesian model to predict a house price for a house with a floor area of 75m^2, providing also a credible interval for this prediction. Repeat the analysis for a house with a floor area of 175m^2.

Using our fitted model from Exercise 2 we calculate the prediction for the price of a house with a floor area of 75 m^2 as having a mean value of €675,327 & with a 90% credible interval between €661,166 and €688,516.

```
# apply f_mu function to log of 75
mu_75 <- sapply(log(75), f_mu)
# calculate highest posterior density interval for the prediction
y_hdi_75 = HDInterval::hdi(mu_75, credMass=0.9)
# calculate mean of the predictions as point estimate of our prediction
mean(mu_75)
```

```
## [1] 675.2142
```

```
# show prediction interval
pi_75 = y_hdi_75[,1]
pi_75
```

```
##    lower    upper
## 661.6245 688.4786
```

Conducting similar analysis to a property with a floor area of 175 m^2, we calculate a mean predicted value of €1,084,152, with a 90% credible interval between €1,070,722 and €1,097,808.

```
# repeat same analysis for area of 175 m^2
mu_175 <- sapply(log(175), f_mu)
y_hdi_175 = HDInterval::hdi(mu_175, credMass=0.9)

pi_175 = y_hdi_175[,1]
mean(mu_175)
```

```
## [1] 1084.294
```

```
pi_175
```

```
##    lower    upper
## 1071.091 1098.188
```