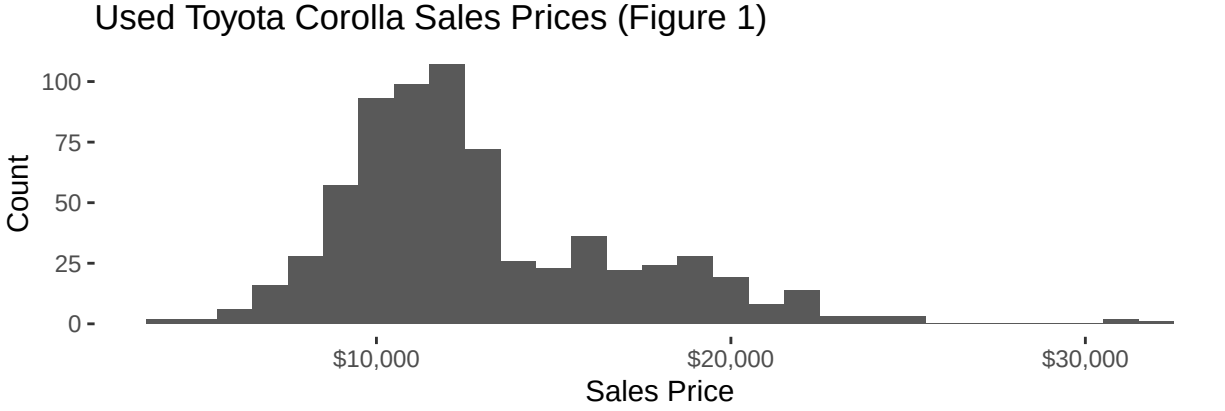
Toyota Case Study

Trevor Isaacson, Runze Jiang, Adam Kiehl

2022-10-12

Introduction

Consumers shopping for new cars often have the option to trade in their current car for money towards their new one. Dealerships, in turn, sell the used cars at a slightly higher price. To do this effectively, the dealership must be able to accurately estimate the expected price that it can sell each used car for based only on characteristics of the car. Specifically, a Toyota dealership was interested in fitting a regression model to estimate `Price` using 16 different predictor variables and a data set of 694 used Toyota Corolla sales. Figure 1 shows the prices of those Corollas and to eliminate the skewness in Price, figure 2 shows the log(price) of those same Corollas.



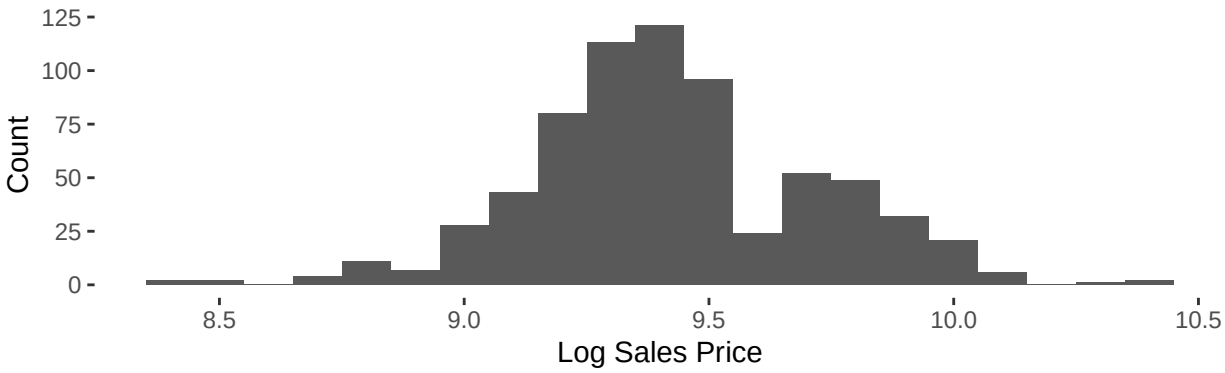


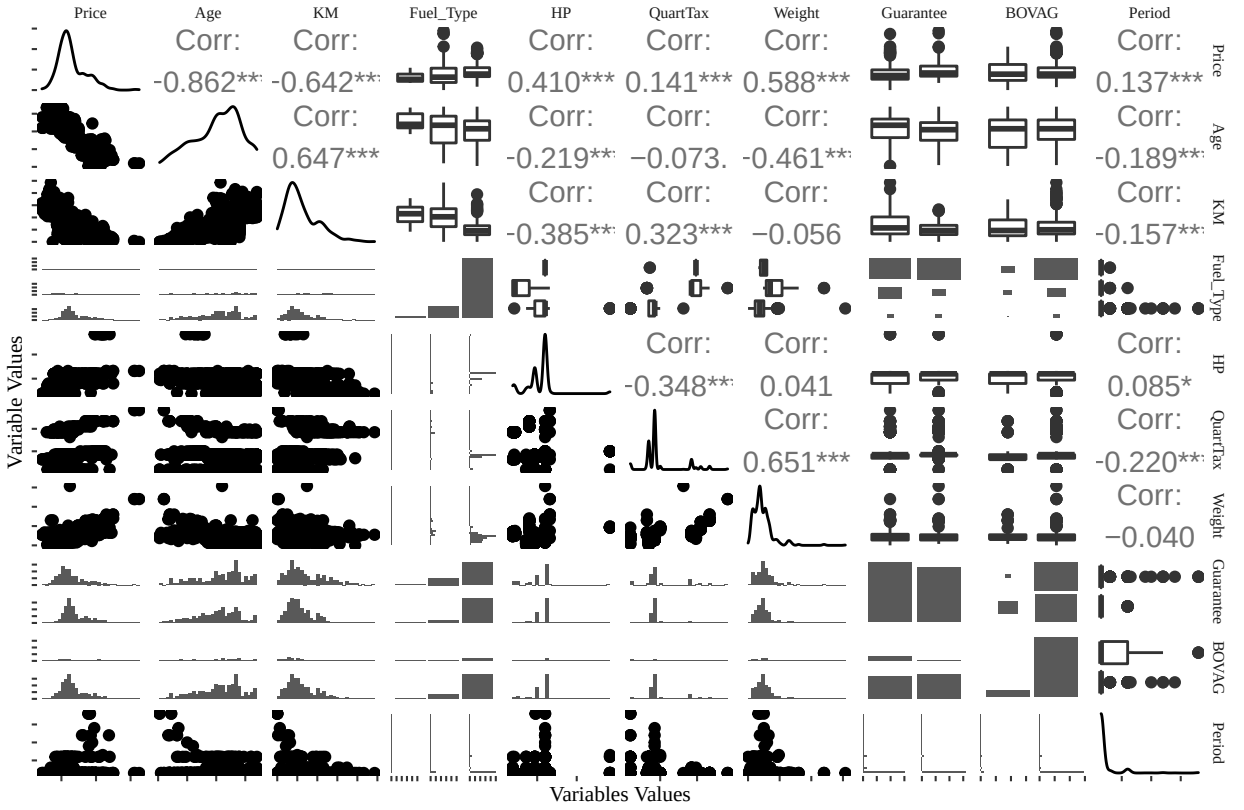
Table 1: Summary Statistics

Variable	Mean	Std. Deviation	Minimum	Maximum
Age	41.31	15.77	1	68
BOVAG	1.90	0.30	1	2
Fuel_Type	2.80	0.44	1	3
Guarantee	1.48	0.50	1	2
HP	102.49	16.95	69	192
KM	60614.31	41309.23	1	243000
Period	4.03	3.81	3	36
Price	12928.45	4012.51	4350	32500
QuartTax	96.47	48.92	19	283
Weight	1090.51	62.13	1000	1615

Numeric predictors included `Age`, `KM` (odometer reading), `HP` (horsepower), `CC` (cylinder volume), `QuartTax` (quarterly road tax), `Weight` and `Period` (guarantee period). Categorical variables included `Mfg_Month` (manufacturing month), `Mfg_Year` (manufacturing year from 1999-2004), `Fuel_type` (CNG, Diesel, or Petrol), `Metallic`, `Automatic`, `Doors` (3, 4, or 5), `Gears` (5 or 6), `Guarantee` (manufacturer's guarantee), and `BOVAG` (BOVAG guarantee). The `Cylinders` predictor was removed from the original data set because all recorded cars had 4 cylinders, and all categorical predictors were made into factors in `R`. No other cleaning was performed on the data.

A set of practical predictors of interest were chosen based on contextual knowledge and exploratory data visualizations to use for model fitting separately of any formal variable selection techniques. Intuitively, older cars (indicated by `Age` and `KM`) should be worth less and larger and more powerful cars (indicated by `Weight` and `HP`, respectively) should be worth more. It was assumed that cars that take different fuel types (`Fuel_Type`) are likely different kinds of cars and would likely vary in price in some way. Finally, `Guarantee`, `BOVAG`, and `Period` are financial-based incentives that should also intuitively lead to a higher car price. Summary statistics (Table 1) and plots (Figure 3) for these predictors of interest are shown below.

Paired Plots and Correlations (Figure 3)



Analysis

The relationships between many of the predictors and the response variable of interest, car price, was largely unknown. Aside from a selection of predictors deemed to likely be important based on contextual knowledge, it was not known which predictors should be included in a regression model. Therefore, an array of models were fit using various transformations, variable selection methods, and prior assumptions. All models were fit with the `stan_glm` function in the `rstanarm` package using a Bayesian framework that incorporates prior belief into into traditional regression by use of prior distributions. After fitting, all models were assessed for explanatory power, overfitting, and violations of model assumptions by comparing R^2 and R^2_{LOO} and by generating QQ-Normal plots for residuals and residual vs fitted values plots.

To begin, a full model (1) was fit using the default `stan_glm` priors ($N(0, 6.25)$) to serve as a comparative baseline for future models. The regression intercept was estimated to be $\beta_0 = -4,173.88$ and can be interpreted as the expected price (in dollars) of 3-door, manual, non-metallic, CNG-fueled car that was manufactured in January of 1999 with no manufacturer or BOVAG guarantee and a value of 0 for all numeric predictors included in the model. The regression coefficients, β_j, could be interpreted as the expected difference in expected car price associated with a one unit difference in the predictor (or associated with whether or not an observation assumes that specific level of a categorical predictor), with all other predictors remaining constant. These interpretations are unreasonable in a practical context and it makes no sense that the price of a car could be estimated to be a negative value. Therefore, another full model (2) was fit, also using the default `stan_glm` priors, but now with scaled numeric predictors and a log-transformation of car price. This was done to make the interpretation of coefficients more practical and comparable, and to restrict predictions of price to positive values. There may also be potential for modeling benefits from scaling. Then, regression coefficients for numeric predictors could be interpreted as the expected percent difference

Table 2: Prior Parameters

Variable	Prior Mean	Prior Std. Deviation
Age	-100	625
BOVAG Guarantee	0	100
Fuel (Diesel)	1000	62500
Fuel (Petrol)	-1000	62500
Guarantee Period	1000	62500
HP	50	100
KM	-0.0625	0.125
Manufacturer Guarantee	1000	62500
Quarterly Tax	100	625
Weight	10	0.125

in expected car price associated with a one standard deviation increase in the value of the predictors, with all other predictors remaining constant.

The full models fitted utilized 32 predictors (16, not including dummy variables) for modeling. Using a large number of predictors can be damaging to a model due to redundancies in predictor information (multicollinearity) and the potential for overfitting. Several different variable selection methods were applied to the car prices data set to determine which predictors were most closely related with car price. First, a Bayesian variable selection method called a horseshoe prior was used to systematically exclude unimportant predictors from the model. A horseshoe prior places a large amount of prior mass near 0, forcing predictors to be highly related with the response to be included in the model. This model (3) determined 14 predictors (12, not including dummy variables) to be significant in predicting car price: `Age`, `Mfg_Month`, `Mfg_Year`, `KM`, `Fuel_Type`, `HP`, `Metallic`, `QuartTax`, `Weight`, `Guarantee`, `BOVAG`, and `Period`. A follow-up model (4) was fit using only these selected predictors, with only an inconsequential loss in R^2. Next, a model (5) was fit using a variable selection method called LASSO that penalizes a model for producing large coefficient estimates, forcing it to be selective with how it chooses estimates (equivalent to using a Laplace prior). This model determined 15 predictors (12, not including dummy variables) to be significant in predicting car price. These were the same predictors chosen by the horseshoe prior model previously fit. A follow-up model (6) was fit using only these selected predictors, again with only an inconsequential loss in R^2.

Thirdly, a more contextual approach was taken to variable selection and the predictors selected previously based on practical knowledge and data visualizations were used for modeling. These predictors were selected independently of the results of the horseshoe and LASSO selections, but aligned closely with these selections nonetheless. First, a model (7) was fit using these predictors and the `stan_glm` default priors. A follow-up model (8) was then fit using using weakly informative priors (Table 2) informed by prior knowledge and exploratory data visualizations. This model showed slightly less evidence of overfitting than the previous model but was otherwise similar. Finally, two models (9) & (10) were fit to explore potential interactions of interest. The first of these models included an interaction between `Age` and `KM`, and the second included an interaction between `Weight` and `HP`.

Results

For our final model, we decided to use model 4 (Table 4). This model was constructed using horse-shoe selected predictors with scaled data and default priors. The horseshoe selected predictors include `Age`, `Mfg_Month`, `Mfg_Year`, `KM`, `Fuel_Type`, `HP`, `Metallic`, `QuartTax`, `Weight`, `Guarantee`, `BOVAG`, and `Period`. Using the scaled numeric predictors and a log-transformation of car price to restrict prediction prices to positive values only, this model was able to better predict selling price compared to other models. Because the goal of this study was price prediction, the final model needed to have great observed predictive powers.

The intercept coefficient can be interpreted as the expected log(price) of a non-metallic, CNG-fueled car

Table 3: Model Fitting Results

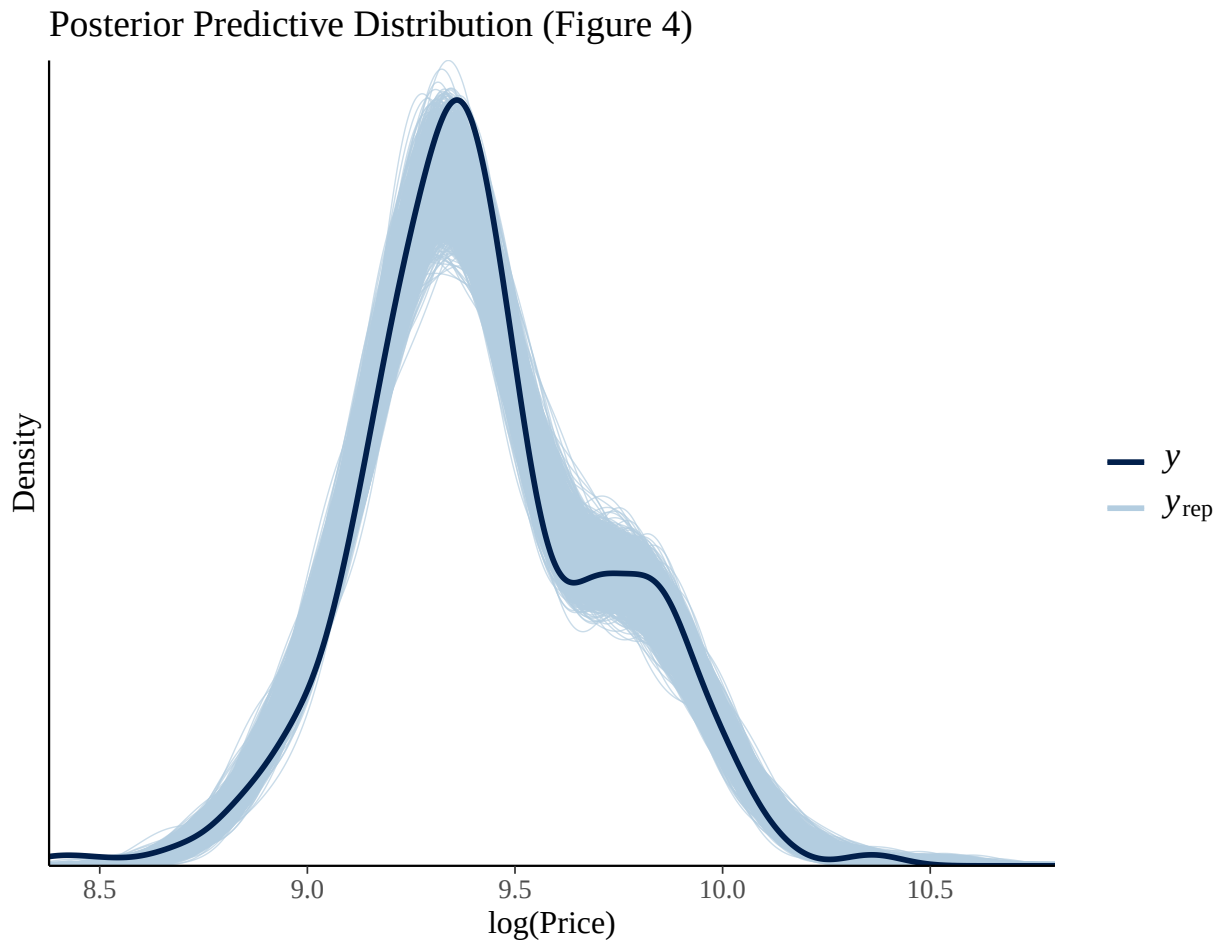
Model	Predictors	R^2	LOO R^2	LOO ELPD
1	32	0.902	0.890	558.92
2	32	0.882	0.871	568.73
3	32	0.880	0.871	574.79
4	27	0.881	0.870	572.49
5	32	0.881	0.870	572.22
6	27	0.881	0.871	574.71
7	10	0.867	0.857	536.79
8	10	0.866	0.863	488.65
9	11	0.867	0.856	537.22
10	12	0.868	0.855	538.42

Table 4: Coefficient Estimates

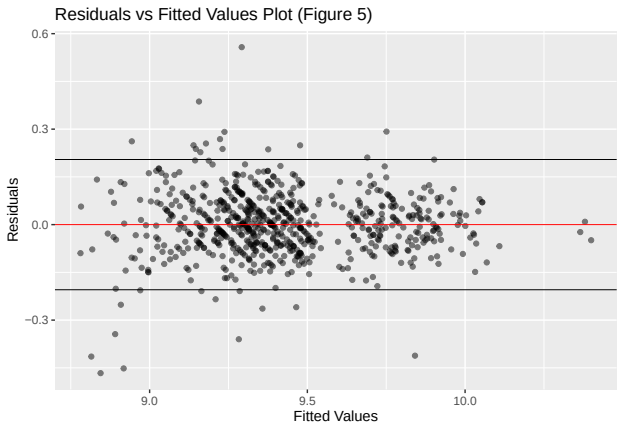
	Estimate	Std. Error
(Intercept)	9.113	0.677
Age	-0.089	0.399
Mfg_Month2	0.025	0.031
Mfg_Month3	0.037	0.053
Mfg_Month4	0.018	0.077
Mfg_Month5	0.043	0.104
Mfg_Month6	0.029	0.128
Mfg_Month7	0.012	0.154
Mfg_Month8	0.027	0.177
Mfg_Month9	-0.024	0.201
Mfg_Month10	-0.012	0.228
Mfg_Month11	-0.007	0.255
Mfg_Month12	0.020	0.282
Mfg_Year2000	0.022	0.305
Mfg_Year2001	0.039	0.611
Mfg_Year2002	0.170	0.913
Mfg_Year2003	0.226	1.217
Mfg_Year2004	0.217	1.517
KM	-0.077	0.006
Fuel_TypeDiesel	0.031	0.034
Fuel_TypePetrol	0.172	0.035
HP	0.050	0.005
Metallic1	0.022	0.009
QuartTax	0.075	0.009
Weight	0.040	0.008
Guarantee1	0.030	0.008
BOVAG1	0.050	0.015
Period	0.015	0.004

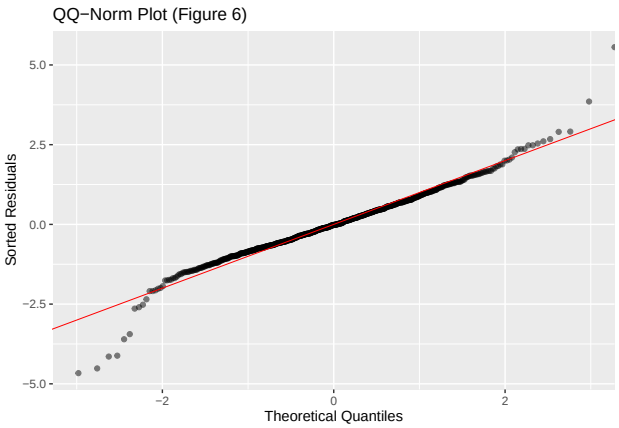
that was manufactured in January of 1999 with no manufacturer or BOVAG guarantee and a value of 0 for all numeric predictors included in the model. The coefficients estimates for the numeric predictors can be interpreted as the expected percent difference in predicted car price associated with a one standard deviation increase in the value of the predictor, with all other predictors remaining constant. This model also produced a residual standard deviation of 0.102. This is a relatively low standard error compared to our model coefficients. Overall, this model had an in-sample Bayesian R^2 value of 0.881 and a leave one out adjusted R^2 value of 0.87. This confirms the model wasn't overfitting while also showing a relatively high R^2 value. Using leave one out cross validation and also k-fold cross validation, all odel's predictive powers were tested. Model 4 was the third best model using leave one out cross validation and the best using k-fold cross validation and based on model and data inputs, thus model 4 as the final model.

Next, the final fitted model should look like our data. By drawing from the predictive distribution and comparing it to the distribution of the response variable (Figure 4), a general assumption can determine if the model is fitting appropriately. In the plot below, we see our predictive distribution tracked the response distribution well increasing our confidence in the model.



Checking the assumptions of our final model, the model included all the relevant predictors as these were chosen using the horseshoe prior thus forcing predictors to be highly related with the response. The outcome measure accurately reflected the prediction interests and is generalized to all Toyota Corollas. Looking at the residual vs fitted values plot (Figure 5), there are no patterns or trends within the residuals vs fitted values plot. Most values are within 2 standard residuals of 0 and the values are spread across the 0 line. There might be some clumping but nothing big enough to question the model. There aren't any heavy tails in the QQ-norm plot (Figure 6) and the values closely align with the red line. These positive assumptions increased the validity of the model.





In all, the purpose of this study was to predict the selling price of used Toyota Corollas and ensure a small profit based on their new purchase and trade-in promotion. Based on several variables, this final model fit will help the dealership closely estimate the final selling price for their used cars. With this model, the dealership can now ensure a reasonable profit by plugging in the characteristics of each individual car and output a predicted selling price. This will result in more accurate selling prices and higher profits for the dealer.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(GGally)
library(rstanarm)
library(ggthemes)
library(scales)
library(knitr)
library(kableExtra)
library(bayesplot)
library(loo)
library(ggplot2)
library(gridExtra)

set.seed(551)
options(scipen=999)
# read data from .csv file
data_raw <- read.csv('ToyotaCorollaData.csv')

data <- data_raw %>%
  # factor categorical predictors
  mutate_at(c('Metallic', 'Automatic', 'Doors', 'Cylinders', 'Gears',
              'Guarantee', 'BOVAG', 'Fuel_Type', 'Mfg_Month', 'Mfg_Year'),
            as.factor) %>%
  # remove singular predictor
  select(-Cylinders)

# log transform Price and scale predictors
data_scale <- cbind(log(data$Price),
                    (data %>%
                       select(-Price) %>%
                       mutate_if(is.numeric, scale))) %>%
  as.data.frame()
names(data_scale) <- names(data)
resultsDF <- data.frame(model = 1:10,
                        predictors = rep(NA, 10),
                        R2 = rep(NA, 10),
                        LOO_R2 = rep(NA, 10),
                        LOO_CV = rep(NA, 10))

# function to assess model and display examination results
diagFit <- function(fit, modelNum, results) {
  # extract number of predictors
  p <- length(fit$coefficients) - 1
  results$predictors[which(results$model == modelNum)] <- p

  # find model R^2 scores
  bayesR2 <- round(mean(bayes_R2(fit)), 3)
  results$R2[which(results$model == modelNum)] <- bayesR2
  looR2 <- round(mean(loo_R2(fit)), 3)
  results$LOO_R2[which(results$model == modelNum)] <- looR2
```

```
  # find model LOO CV score
  looFit <- loo(fit, k_threshold = 0.7)
  elpd <- round(looFit$estimates[1, 1], 2)
  # correction for log transformation of response
  if (elpd < 0) {
    looFit$pointwise[, 1] <- looFit$pointwise[, 1] + data_scale$Price
    elpd <- round(sum(looFit$pointwise[, 1]), 2)
  }
  results$LOO_CV[which(results$model == modelNum)] <- elpd

  return(results)
}

N <- 1000

fit1 <- stan_glm(Price ~ .,
                 data = data,
                 refresh = 0,
                 iter = N)
resultsDF <- diagFit(fit1, 1, resultsDF)

fit2 <- stan_glm(Price ~ .,
                 data = data_scale,
                 refresh = 0,
                 iter = N)
resultsDF <- diagFit(fit2, 2, resultsDF)

p <- ncol(data) - 1
n <- nrow(data)

p0 <- 6

slab_scale <- sqrt(0.3 / p0) * sd(data_scale$Price)
global_scale <- (p0 / (p - p0)) / sqrt(n)

fit3 <- stan_glm(Price ~ .,
                 data = data_scale,
                 refresh = 0,
                 iter = N,
                 prior = hs(global_scale = global_scale,
                            slab_scale = slab_scale))
resultsDF <- diagFit(fit3, 3, resultsDF)

fit4 <- stan_glm(Price ~ Age + Mfg_Month + Mfg_Year + KM + Fuel_Type + HP +
                   Metallic + QuartTax + Weight + Guarantee + BOVAG + Period,
                 data = data_scale,
                 refresh = 0,
                 iter = N)
resultsDF <- diagFit(fit4, 4, resultsDF)

fit5 <- stan_glm(Price ~ .,
                 data = data_scale,
                 refresh = 0,
```

```
                  iter = N,
                  prior = lasso())
resultsDF <- diagFit(fit5, 5, resultsDF)

fit6 <- stan_glm(Price ~ Age + Mfg_Month + Mfg_Year + KM + Fuel_Type + HP +
                   Metallic + QuartTax + Weight + Guarantee + BOVAG + Period,
                 data = data_scale,
                 refresh = 0,
                 iter = N)
resultsDF <- diagFit(fit6, 6, resultsDF)

fit7 <- stan_glm(Price ~ Age + KM + Fuel_Type + HP + QuartTax + Weight +
                   Guarantee + BOVAG + Period,
                 data = data_scale,
                 refresh = 0,
                 iter = N)
resultsDF <- diagFit(fit7, 7, resultsDF)

priorMeans <- c(-100, -.0625, 1000, -1000, 50, 100, 10, 1000, 0, 1000)
priorVars <- c(625, .125, 62500, 62500, 100, 625, .125, 62500, 100, 62500)

fit8 <- stan_glm(Price ~ Age + KM + Fuel_Type + HP + QuartTax + Weight +
                   Guarantee + BOVAG + Period,
                 data = data,
                 refresh = 0,
                 iter = N,
                 prior = normal(priorMeans,
                                priorVars))
resultsDF <- diagFit(fit8, 8, resultsDF)

fit9 <- stan_glm(Price ~ Fuel_Type + HP + QuartTax + Weight +
                   Guarantee + BOVAG + Period + Age*KM,
                 data = data_scale,
                 refresh = 0,
                 iter = N)
resultsDF <- diagFit(fit9, 9, resultsDF)

fit10 <- stan_glm(Price ~ Fuel_Type + HP + QuartTax + Weight +
                    Guarantee + BOVAG + Period + Age*KM + Weight*HP,
                  data = data_scale,
                  refresh = 0,
                  iter = N)
resultsDF <- diagFit(fit10, 10, resultsDF)
plt1 <- ggplot(data,
               mapping = aes(x = Price)) +
  geom_histogram(binwidth = 1000) +
  theme_tufte(base_family = 'sans') +
  scale_x_continuous(labels = label_dollar(prefix = '$')) +
  scale_y_continuous(breaks = seq(0, 125, by = 25)) +
  labs(title = 'Used Toyota Corolla Sales Prices (Figure 1)',
       x = 'Sales Price',
       y = 'Count')
```

```
plt2 <- ggplot(data_scale,
               mapping = aes(x = Price)) +
  geom_histogram(binwidth = .1) +
  theme_tufte(base_family = 'sans') +
  scale_y_continuous(breaks = seq(0, 125, by = 25)) +
  labs(title = 'Used Toyota Corolla Log Sales Prices (Figure 2)',
       x = 'Log Sales Price',
       y = 'Count')

grid.arrange(plt1, plt2)
data %>%
  select(c(Price, Age, KM, Fuel_Type, HP, QuartTax, Weight, Guarantee, BOVAG,
           Period)) %>%
  mutate_if(is.factor, as.numeric) %>%
  pivot_longer(c(Price, Age, KM, Fuel_Type, HP, QuartTax, Weight, Guarantee,
                 BOVAG, Period),
               names_to = 'Variable',
               values_to = 'Value') %>%
  group_by(Variable) %>%
  summarize(Mean = mean(Value) %>%
              round(2),
            'Std. Deviation' = sd(Value) %>%
              round(2),
            Minimum = min(Value),
            Maximum = max(Value)) %>%
  kbl(align = c('l', 'r', 'r', 'r', 'r'),
      caption = 'Summary Statistics') %>%
  kable_classic()

data %>%
  select(c(Price, Age, KM, Fuel_Type, HP, QuartTax, Weight, Guarantee, BOVAG,
           Period)) %>%
  ggpairs(progress = FALSE) +
  theme_tufte(base_size = 8) +
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank()) +
  labs(title = 'Paired Plots and Correlations (Figure 3)',
       x = 'Variables Values',
       y = 'Variable Values')
data.frame(Variable = c('Age', 'KM', 'Fuel (Diesel)', 'Fuel (Petrol)', 'HP',
                        'Quarterly Tax', 'Weight', 'Manufacturer Guarantee',
                        'BOVAG Guarantee', 'Guarantee Period'),
           PriorMean = as.character(c(-100, -0.0625, 1000, -1000, 50, 100, 10,
                                      1000, 0, 1000)),
           PriorStdDev = as.character(c(625, 0.125, 62500, 62500, 100, 625,
                                        0.125, 62500, 100, 62500))) %>%
  mutate('Prior Mean' = PriorMean,
         'Prior Std. Deviation' = PriorStdDev) %>%
  select(-c(PriorMean, PriorStdDev)) %>%
  arrange(Variable) %>%
  kbl(align = c('l', 'r', 'r'),
      caption = 'Prior Parameters') %>%
  kable_classic()
```

```
resultsDF %>%
  kbl(col.names = c('Model', 'Predictors', 'R^2', 'LOO R^2', 'LOO ELPD'),
      align = c('l', rep('r', 4)),
      caption = 'Model Fitting Results') %>%
  kable_classic()
data.frame('Estimate' = round(fit4$coefficients, 3),
           'Std. Error' = round(fit4$ses, 3)) %>%
  kbl(col.names = c('Estimate', 'Std. Error'),
      align = c('r', 'r'),
      caption = 'Coefficient Estimates') %>%
  kable_classic()
fit4_rep = posterior_predict(fit4)
ppc_dens_overlay(data_scale$Price, fit4_rep) +
  scale_y_continuous(breaks=NULL) +
  labs(title = 'Posterior Predictive Distribution (Figure 4)',
       x = 'log(Price)',
       y = 'Density')
plt_res_fit =ggplot(mapping = aes(x = fit4$fitted.values, y = fit4$residuals)) +
    geom_point(alpha = .5) +
    geom_hline(yintercept = 0, col = 'red', size = .2) +
    geom_hline(yintercept = 2*sigma(fit4), size = 0.1) +
    geom_hline(yintercept = -2*sigma(fit4), size = 0.1) +
    labs(title = 'Residuals vs Fitted Values Plot (Figure 5)',
         x = 'Fitted Values',
         y = 'Residuals')

n <- length(fit4$residuals)
quants <- qnorm((1:n / n))
plt_QQ <- ggplot(mapping = aes(x = quants,  y = sort(scale(fit4$residuals)))) +
  geom_point(alpha = .5) +
  geom_abline(intercept = 0,
              slope = 1,
              col = 'red',
              size = .2) +
  labs(title = 'QQ-Norm Plot (Figure 6)',
       x = 'Theoretical Quantiles',
       y = 'Sorted Residuals')

plt_res_fit
plt_QQ
```