

Fitted Models

Adam Kiehl

2022-10-11

```
set.seed(551)
```

Import Data

```
# 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)

head(data)
```

```
##   Price Age Mfg_Month Mfg_Year    KM Fuel_Type HP Metallic Automatic   CC Doors
## 1 13500  23         10     2002 46986   Diesel 90         1           0 2000     3
## 2 13750  23         10     2002 72937   Diesel 90         1           0 2000     3
## 3 13950  24          9     2002 41711   Diesel 90         1           0 2000     3
## 4 14950  26          7     2002 48000   Diesel 90         0           0 2000     3
## 5 13750  30          3     2002 38500   Diesel 90         0           0 2000     3
## 6 12950  32          1     2002 61000   Diesel 90         0           0 2000     3
##   Gears QuartTax Weight Guarantee BOVAG Period
## 1     5      210  1165          0     1     3
## 2     5      210  1165          0     1     3
## 3     5      210  1165          1     1     3
## 4     5      210  1165          1     1     3
## 5     5      210  1170          1     1     3
## 6     5      210  1170          0     1     3
```

Scale data and log transform Price.

```
# 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)
```

Diagnostic Function

```
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, printPlots) {
  # print model fit
  print(fit,
        digits = 3,
        detail = FALSE)

  # extract number of predictors
  p <- length(fit$coefficients) - 1
  results$predictors[which(results$model == modelNum)] <- p

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

  # print mode 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
  print(paste('LOO ELPD: ', elpd))

  # generate QQ-Norm plot
  n <- length(fit$residuals)
  quants <- qnorm((1:n / n))

  plt_QQ <- ggplot(mapping = aes(x = quants,
                                y = sort(scale(fit$residuals)))) +
    geom_point(alpha = .5) +
    geom_abline(intercept = 0,
                slope = 1,
                col = 'red',
```

```

        size = .2) +
  labs(title = 'QQ-Norm Plot',
        x = 'Theoretical Quantiles',
        y = 'Sorted Residuals')

# generate residuals vs fitted values plot
plt_res_fit <- ggplot(mapping = aes(x = fit$fitted.values,
                                    y = fit$residuals)) +

  geom_point(alpha = .5) +
  geom_hline(yintercept = 0,
            col = 'red',
            size = .2) +
  labs(title = 'Residuals vs Fitted Values Plot',
        x = 'Fitted Values',
        y = 'Residuals')

# print plots if desired
if (printPlots) {
  print(plt_QQ)
  print(plt_res_fit)
}

return(results)
}

# Bayesian simulation size
N <- 5000

# print model diagnostic plots?
printPlots <- FALSE

```

Model 1: Full, Unscaled, Default Priors

```

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

resultsDF <- diagFit(fit1, 1, resultsDF, printPlots)

```

##	Median	MAD_SD
## (Intercept)	-4173.885	23006.583
## Age	-84.341	336.838
## Mfg_Month2	205.219	384.232
## Mfg_Month3	348.234	702.700
## Mfg_Month4	146.004	1022.252
## Mfg_Month5	506.995	1357.477
## Mfg_Month6	211.164	1692.619
## Mfg_Month7	-85.699	2014.251
## Mfg_Month8	90.152	2359.808
## Mfg_Month9	-383.527	2704.662

```

## Mfg_Month10      -325.799  3022.337
## Mfg_Month11      -351.001  3380.437
## Mfg_Month12      -145.222  3683.105
## Mfg_Year2000      -185.908  4036.807
## Mfg_Year2001      -314.582  8079.371
## Mfg_Year2002      1617.607 12158.539
## Mfg_Year2003      2735.279 16201.595
## Mfg_Year2004      3915.591 20215.338
## KM                -0.020    0.002
## Fuel_TypeDiesel   1075.313   414.333
## Fuel_TypePetrol   1688.934   438.442
## HP                50.020    4.657
## Metallic1         179.532   109.084
## Automatic1        425.403   254.684
## CC                 0.075    0.092
## Doors4            -25.385   202.744
## Doors5             90.795   115.540
## Gears6             95.111   339.885
## QuartTax           13.535    2.314
## Weight             11.324    1.706
## Guarantee1        370.747   107.178
## BOVAG1             410.389   189.980
## Period             25.924    14.765
##
## Auxiliary parameter(s):
##      Median  MAD_SD
## sigma 1258.207  34.772
## [1] "R^2:  0.902"
## [1] "LOO R^2:  0.89"
## [1] "LOO ELPD: 563.82"

```

Model 2: Full, Scaled, Default Priors

```

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

resultsDF <- diagFit(fit2, 2, resultsDF, printPlots)

```

```

##      Median MAD_SD
## (Intercept)   9.109 0.670
## Age          -0.086 0.396
## Mfg_Month2     0.024 0.030
## Mfg_Month3     0.038 0.053
## Mfg_Month4     0.018 0.078
## Mfg_Month5     0.041 0.102
## Mfg_Month6     0.027 0.127
## Mfg_Month7     0.006 0.152
## Mfg_Month8     0.024 0.177
## Mfg_Month9    -0.025 0.203
## Mfg_Month10   -0.009 0.228

```

```

## Mfg_Month11      -0.010  0.255
## Mfg_Month12       0.016  0.278
## Mfg_Year2000       0.025  0.303
## Mfg_Year2001       0.038  0.602
## Mfg_Year2002       0.181  0.906
## Mfg_Year2003       0.237  1.202
## Mfg_Year2004       0.235  1.506
## KM                -0.078  0.007
## Fuel_TypeDiesel    0.037  0.034
## Fuel_TypePetrol    0.162  0.034
## HP                 0.051  0.006
## Metallic1          0.021  0.009
## Automatic1         0.034  0.021
## CC                 0.004  0.004
## Doors4             -0.007  0.016
## Doors5             0.016  0.009
## Gears6             0.015  0.028
## QuartTax           0.076  0.009
## Weight             0.032  0.008
## Guarantee1         0.030  0.009
## BOVAG1             0.054  0.015
## Period             0.016  0.005
##
## Auxiliary parameter(s):
##      Median MAD_SD
## sigma 0.102  0.003
## [1] "R^2:  0.882"
## [1] "LOO R^2:  0.87"
## [1] "LOO ELPD: 568.85"

```

Model 3: Full, Scaled, Horseshoe Prior

```

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))

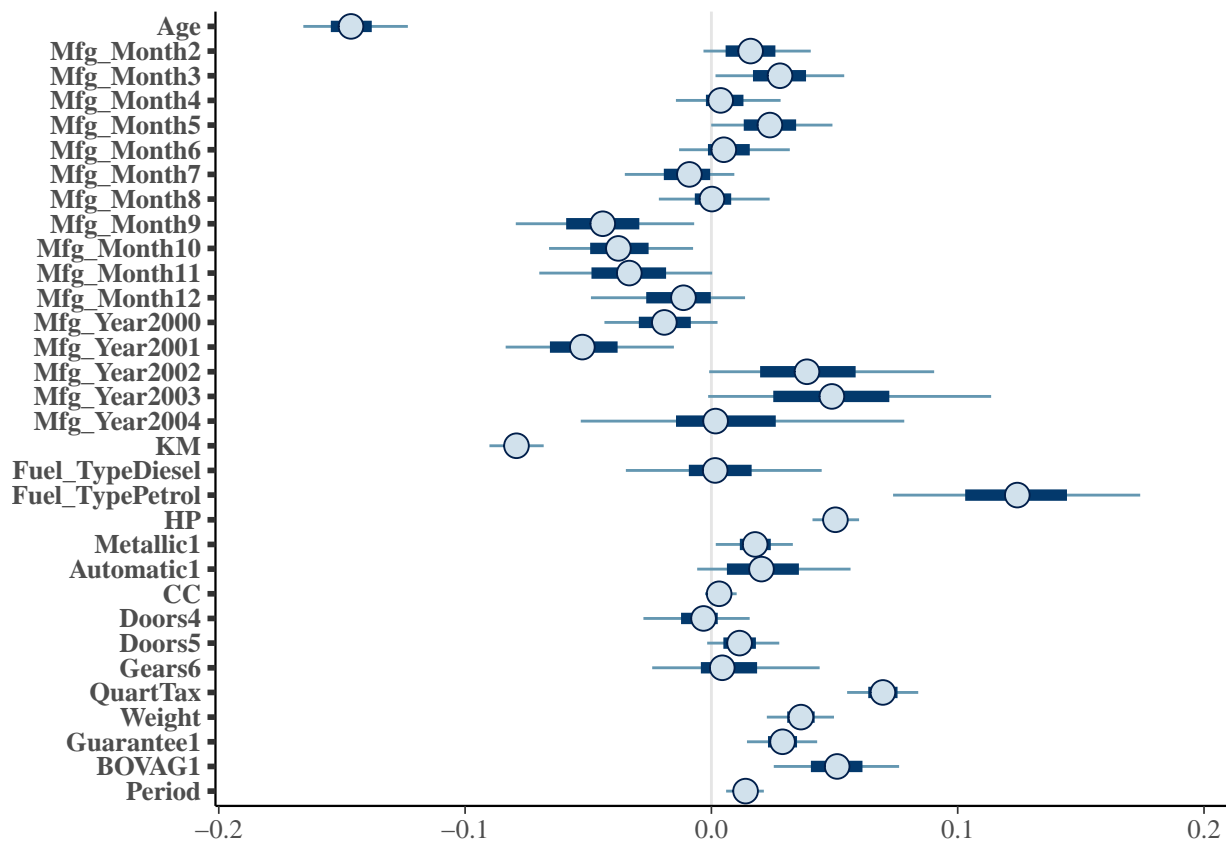
resultsDF <- diagFit(fit3, 3, resultsDF, printPlots)

##      Median MAD_SD
## (Intercept)   9.251  0.036
## Age          -0.146  0.012

```

```
## Mfg_Month2      0.016  0.015
## Mfg_Month3      0.028  0.016
## Mfg_Month4      0.004  0.011
## Mfg_Month5      0.024  0.016
## Mfg_Month6      0.005  0.012
## Mfg_Month7     -0.009  0.013
## Mfg_Month8      0.000  0.011
## Mfg_Month9     -0.044  0.022
## Mfg_Month10    -0.038  0.018
## Mfg_Month11    -0.033  0.022
## Mfg_Month12    -0.011  0.018
## Mfg_Year2000   -0.019  0.016
## Mfg_Year2001   -0.052  0.020
## Mfg_Year2002    0.039  0.029
## Mfg_Year2003    0.049  0.035
## Mfg_Year2004    0.002  0.030
## KM             -0.079  0.007
## Fuel_TypeDiesel 0.002  0.019
## Fuel_TypePetrol 0.124  0.031
## HP             0.050  0.006
## Metallic1      0.018  0.009
## Automatic1     0.020  0.022
## CC             0.003  0.004
## Doors4        -0.003  0.011
## Doors5         0.011  0.010
## Gears6         0.004  0.016
## QuartTax       0.070  0.009
## Weight         0.036  0.008
## Guarantee1     0.029  0.009
## BOVAG1         0.051  0.015
## Period         0.014  0.005
##
## Auxiliary parameter(s):
##      Median MAD_SD
## sigma 0.102  0.003
## [1] "R^2:  0.88"
## [1] "LOO R^2:  0.87"
## [1] "LOO ELPD:  572.39"
```

```
as.data.frame(fit3) %>%
  select(-c('(Intercept)', 'sigma')) %>%
  mcmc_intervals()
```



Model 4: Horseshoe-Selected, Scaled, Default Priors

```
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, printPlots)
```

##	Median	MAD_SD
## (Intercept)	9.169	0.653
## Age	-0.120	0.383
## Mfg_Month2	0.022	0.029
## Mfg_Month3	0.034	0.052
## Mfg_Month4	0.012	0.075
## Mfg_Month5	0.033	0.100
## Mfg_Month6	0.017	0.123
## Mfg_Month7	-0.003	0.148
## Mfg_Month8	0.014	0.172
## Mfg_Month9	-0.040	0.197
## Mfg_Month10	-0.029	0.219
## Mfg_Month11	-0.029	0.245
## Mfg_Month12	-0.004	0.268
## Mfg_Year2000	-0.004	0.293

```
## Mfg_Year2001      -0.016  0.585
## Mfg_Year2002       0.096  0.874
## Mfg_Year2003       0.127  1.164
## Mfg_Year2004       0.094  1.453
## KM                -0.078  0.007
## Fuel_TypeDiesel    0.030  0.034
## Fuel_TypePetrol    0.171  0.035
## HP                 0.050  0.006
## Metallic1          0.022  0.009
## QuartTax           0.076  0.009
## Weight             0.040  0.008
## Guarantee1         0.029  0.009
## BOVAG1             0.050  0.015
## Period             0.015  0.005
##
## Auxiliary parameter(s):
##      Median MAD_SD
## sigma 0.102  0.003
## [1] "R^2:  0.881"
## [1] "LOO R^2:  0.869"
## [1] "LOO ELPD: 573.45"
```

Model 5: Full, Scaled, LASSO Prior

```
fit5 <- stan_glm(Price ~ .,
                 data = data_scale,
                 refresh = 0,
                 iter = N,
                 prior = lasso())

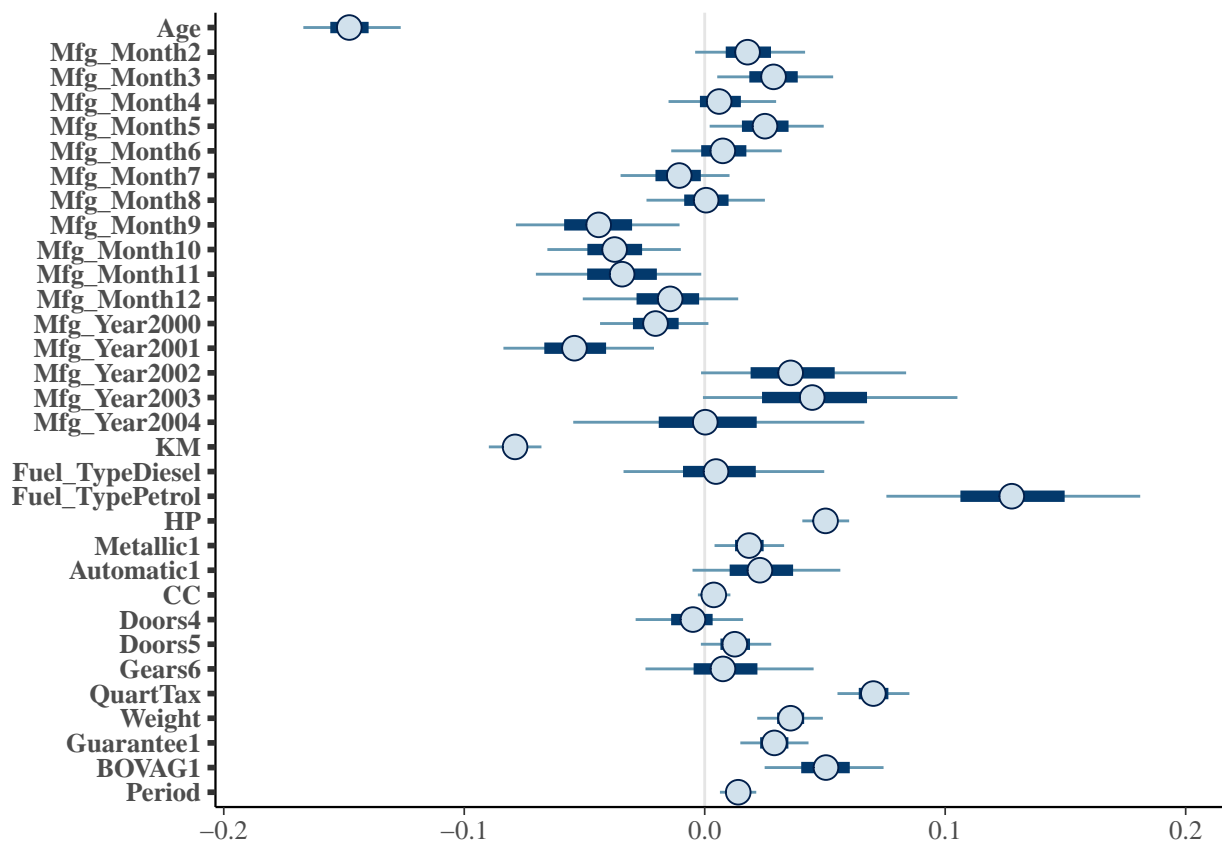
resultsDF <- diagFit(fit5, 5, resultsDF, printPlots)
```

```
##      Median MAD_SD
## (Intercept)    9.249  0.036
## Age          -0.148  0.012
## Mfg_Month2     0.018  0.014
## Mfg_Month3     0.029  0.015
## Mfg_Month4     0.006  0.013
## Mfg_Month5     0.025  0.014
## Mfg_Month6     0.008  0.014
## Mfg_Month7    -0.011  0.014
## Mfg_Month8     0.001  0.014
## Mfg_Month9    -0.044  0.021
## Mfg_Month10   -0.037  0.017
## Mfg_Month11   -0.034  0.021
## Mfg_Month12   -0.014  0.019
## Mfg_Year2000  -0.020  0.014
## Mfg_Year2001  -0.054  0.019
## Mfg_Year2002   0.036  0.026
## Mfg_Year2003   0.045  0.032
## Mfg_Year2004   0.000  0.030
## KM            -0.079  0.007
```



```
## Fuel_TypeDiesel 0.005 0.022
## Fuel_TypePetrol 0.128 0.032
## HP 0.050 0.006
## Metallic1 0.018 0.009
## Automatic1 0.023 0.019
## CC 0.004 0.004
## Doors4 -0.005 0.013
## Doors5 0.013 0.009
## Gears6 0.008 0.020
## QuartTax 0.070 0.009
## Weight 0.036 0.008
## Guarantee1 0.029 0.009
## BOVAG1 0.050 0.015
## Period 0.014 0.005
##
## Auxiliary parameter(s):
## Median MAD_SD
## sigma 0.102 0.003
## [1] "R^2: 0.881"
## [1] "LOO R^2: 0.871"
## [1] "LOO ELPD: 572.17"
```

```
as.data.frame(fit5) %>%
  select(-c('Intercept', 'sigma')) %>%
  mcmc_intervals()
```



Model 6: LASSO-Selected, Scaled, Default Priors

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

resultsDF <- diagFit(fit6, 6, resultsDF, printPlots)
```

```
##           Median MAD_SD
## (Intercept)    9.202  0.656
## Age          -0.130  0.384
## Mfg_Month2     0.022  0.029
## Mfg_Month3     0.035  0.051
## Mfg_Month4     0.009  0.075
## Mfg_Month5     0.032  0.099
## Mfg_Month6     0.014  0.123
## Mfg_Month7    -0.006  0.147
## Mfg_Month8     0.008  0.172
## Mfg_Month9    -0.043  0.197
## Mfg_Month10   -0.032  0.220
## Mfg_Month11   -0.035  0.246
## Mfg_Month12   -0.009  0.269
## Mfg_Year2000  -0.009  0.291
## Mfg_Year2001  -0.026  0.584
## Mfg_Year2002   0.077  0.876
## Mfg_Year2003   0.104  1.168
## Mfg_Year2004   0.068  1.459
## KM           -0.078  0.007
## Fuel_TypeDiesel 0.026  0.034
## Fuel_TypePetrol 0.166  0.036
## HP            0.049  0.006
## QuartTax       0.075  0.009
## Weight         0.041  0.008
## Guarantee1     0.033  0.009
## BOVAG1         0.049  0.015
## Period        0.015  0.005
##
## Auxiliary parameter(s):
##           Median MAD_SD
## sigma 0.103  0.003
## [1] "R^2: 0.88"
## [1] "LOO R^2: 0.87"
## [1] "LOO ELPD: 571.44"
```

Model 7: Selective, Scaled, Default Priors

```
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, printPlots)

```

```

##              Median MAD_SD
## (Intercept)    9.198  0.036
## Age           -0.155  0.007
## KM            -0.071  0.007
## Fuel_TypeDiesel 0.019  0.036
## Fuel_TypePetrol 0.195  0.037
## HP            0.047  0.006
## QuartTax       0.073  0.010
## Weight         0.061  0.008
## Guarantee1     0.028  0.009
## BOVAG1         0.053  0.016
## Period         0.017  0.005
##
## Auxiliary parameter(s):
##           Median MAD_SD
## sigma 0.108  0.003
## [1] "R^2:  0.867"
## [1] "LOO R^2:  0.857"
## [1] "LOO ELPD: 538.93"

```

Model 8: Selective, Unscaled, Weakly Informative Priors

```

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, printPlots)

```

```

##              Median   MAD_SD
## (Intercept)  -870.040 803.751
## Age          -151.073  5.184
## KM           -0.016  0.002
## Fuel_TypeDiesel 1537.554 462.882
## Fuel_TypePetrol 2401.225 490.257
## HP           56.172  4.155
## QuartTax      18.417  2.468
## Weight        10.054  0.128
## Guarantee1    274.489 118.584

```

```
## BOVAG1          62.257   91.575
## Period          33.799   16.142
##
## Auxiliary parameter(s):
##      Median   MAD_SD
## sigma 1463.010   39.630
## [1] "R^2:  0.865"
## [1] "LOO R^2:  0.863"
## [1] "LOO ELPD: 488.67"
```

Model 9: Selective with 1 Interaction, Scaled, Default Priors

```
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, printPlots)
```

```
##              Median MAD_SD
## (Intercept)   9.194  0.035
## Fuel_TypeDiesel 0.023  0.035
## Fuel_TypePetrol 0.195  0.036
## HP             0.048  0.006
## QuartTax       0.074  0.009
## Weight         0.059  0.008
## Guarantee1     0.029  0.009
## BOVAG1         0.053  0.015
## Period         0.015  0.005
## Age            -0.153  0.007
## KM             -0.075  0.007
## Age:KM         0.007  0.005
##
## Auxiliary parameter(s):
##      Median MAD_SD
## sigma 0.108  0.003
## [1] "R^2:  0.867"
## [1] "LOO R^2:  0.857"
## [1] "LOO ELPD: 539.57"
```

Model 10: Selective with 2 Interactions, Scaled, Default Priors

```
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, printPlots)
```

```
##               Median MAD_SD
## (Intercept)    9.189  0.036
## Fuel_TypeDiesel 0.030  0.036
## Fuel_TypePetrol 0.199  0.036
## HP              0.057  0.007
## QuartTax        0.074  0.009
## Weight          0.060  0.008
## Guarantee1      0.028  0.009
## BOVAG1          0.053  0.016
## Period          0.015  0.005
## Age             -0.152  0.007
## KM              -0.076  0.007
## Age:KM           0.008  0.005
## HP:Weight        -0.008  0.005
##
## Auxiliary parameter(s):
##           Median MAD_SD
## sigma 0.108  0.003
## [1] "R^2:  0.867"
## [1] "LOO R^2:  0.857"
## [1] "LOO ELPD: 539.63"
```

Results

```
resultsDF
```

```
##   model predictors      R2 LOO_R2 LOO_CV
## 1      1           32 0.902  0.890 563.82
## 2      2           32 0.882  0.870 568.85
## 3      3           32 0.880  0.870 572.39
## 4      4           27 0.881  0.869 573.45
## 5      5           32 0.881  0.871 572.17
## 6      6           26 0.880  0.870 571.44
## 7      7           10 0.867  0.857 538.93
## 8      8           10 0.865  0.863 488.67
## 9      9           11 0.867  0.857 539.57
## 10    10           12 0.867  0.857 539.63
```

```
loo1 <- loo(fit1, k_threshold = 0.7)
loo2 <- loo(fit2, k_threshold = 0.7)
loo3 <- loo(fit3, k_threshold = 0.7)
loo4 <- loo(fit4, k_threshold = 0.7)
loo5 <- loo(fit5, k_threshold = 0.7)
loo6 <- loo(fit6, k_threshold = 0.7)
loo7 <- loo(fit7, k_threshold = 0.7)
loo8 <- loo(fit8, k_threshold = 0.7)
loo9 <- loo(fit9, k_threshold = 0.7)
loo10 <- loo(fit10, k_threshold = 0.7)

loo1$pointwise[, 1] <- loo1$pointwise[, 1] + data_scale$Price
loo8$pointwise[, 1] <- loo8$pointwise[, 1] + data_scale$Price
```

```
loo_compare(loo1, loo2, loo3, loo4, loo5, loo6, loo7, loo8, loo9, loo10)
```

```
##          elpd_diff se_diff
## fit4      0.0         0.0
## fit3     -0.9         4.7
## fit5     -1.7         4.2
## fit6     -2.4         2.7
## fit2     -5.3         5.6
## fit9    -33.3        13.1
## fit10   -35.2        13.9
## fit7    -35.9        16.1
## fit1    -14.0        26.5
## fit8   -85.5        23.1
```