Inflation, Unemployment, Mortgage, Stocks, Population: Evaluating the Sensitivity of the Housing Market

Will Gillette

Set-Up

```
# Loading the Required Packages
library(tidyverse)
library(rstanarm)
library(GGally)
library(gridExtra)
library(bayesplot)
```

```
# Loading the annual housing data set
data.annual <- read_csv('housing/Annual_Macroeconomic_Factors.csv')
```

```
## Rows: 47 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (8): House_Price_Index, Stock_Price_Index, Consumer_Price_Index, Popula...
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Analysis

Fitting the First Analysis Model

```
# Using the same model from the First Analysis
fa.model <- stan_glm(House_Price_Index~
                       Date+
```

```r
                        Stock_Price_Index+
                        Inflation_Rate+
                        Mortgage_Rate+
                        Unemployment_Rate+
                        Population_Growth_Rate,
                      data=data.annual,
                      refresh=0)
fa.model
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      House_Price_Index ~ Date + Stock_Price_Index + Inflation_Rate +
##     Mortgage_Rate + Unemployment_Rate + Population_Growth_Rate
##  observations: 47
##  predictors:   7
## ------
##                        Median   MAD_SD
## (Intercept)            -13158.2   2608.1
## Date                        6.7      1.3
## Stock_Price_Index           0.0      0.0
## Inflation_Rate              2.6      2.1
## Mortgage_Rate               1.2      2.2
## Unemployment_Rate          -2.4      3.0
## Population_Growth_Rate    -51.1     29.0
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 23.9    2.7
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Evaluating Assumptions of the FA Model

```r
# Displaying the residual plot for this baseline model
fa.fitted <- fitted(fa.model)
fa.resid <- resid(fa.model)
ggplot(data.annual, aes(x=fa.fitted,y=fa.resid)) +
  geom_point() +
  geom_abline(slope=0,color='red') +
  labs(title="Baseline Model\nResidual Plot",
```

```
      x="Fitted HPI Values",
      y="Residuals")
```

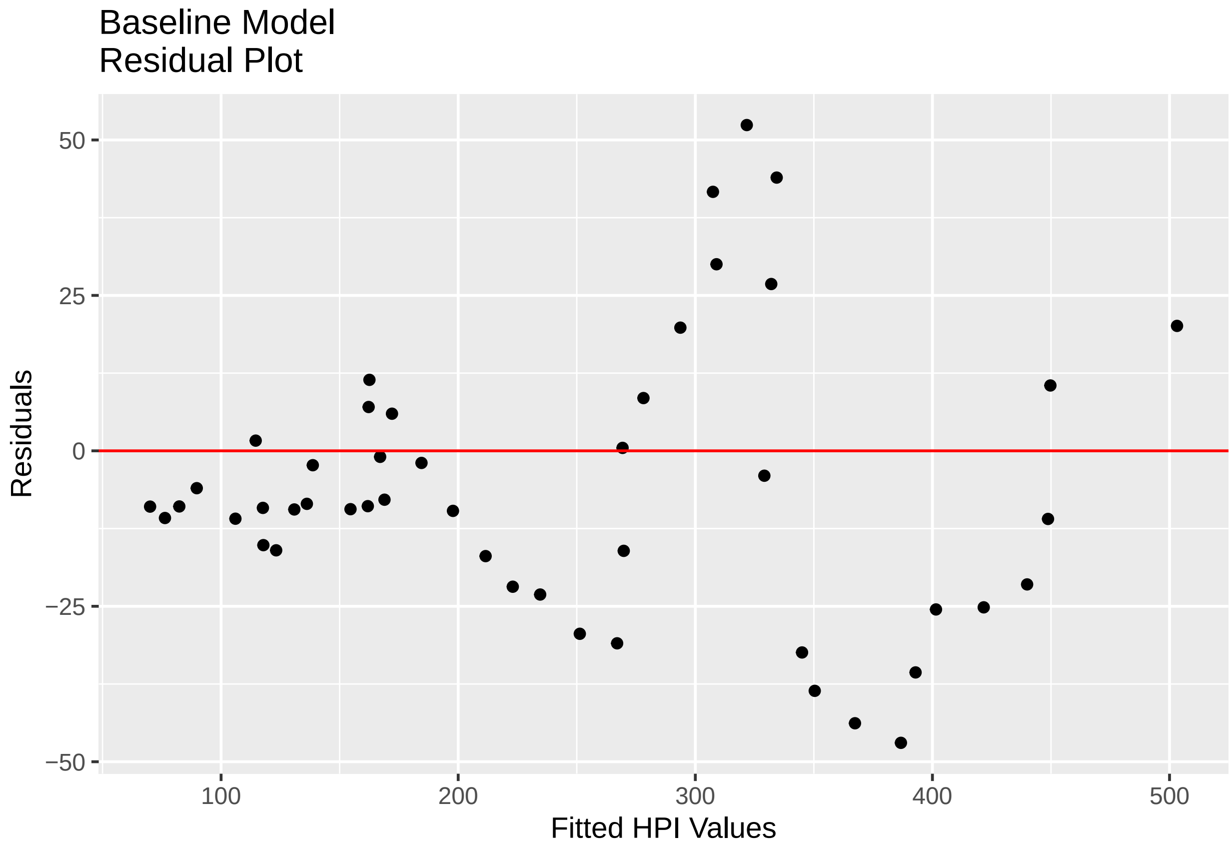

Figure 1.1: A residual vs. fitted plot for the baseline model.

```
fa.stand.resid <- fa.resid/sd(fa.resid)
qqnorm(fa.stand.resid)
qqline(fa.stand.resid)
```

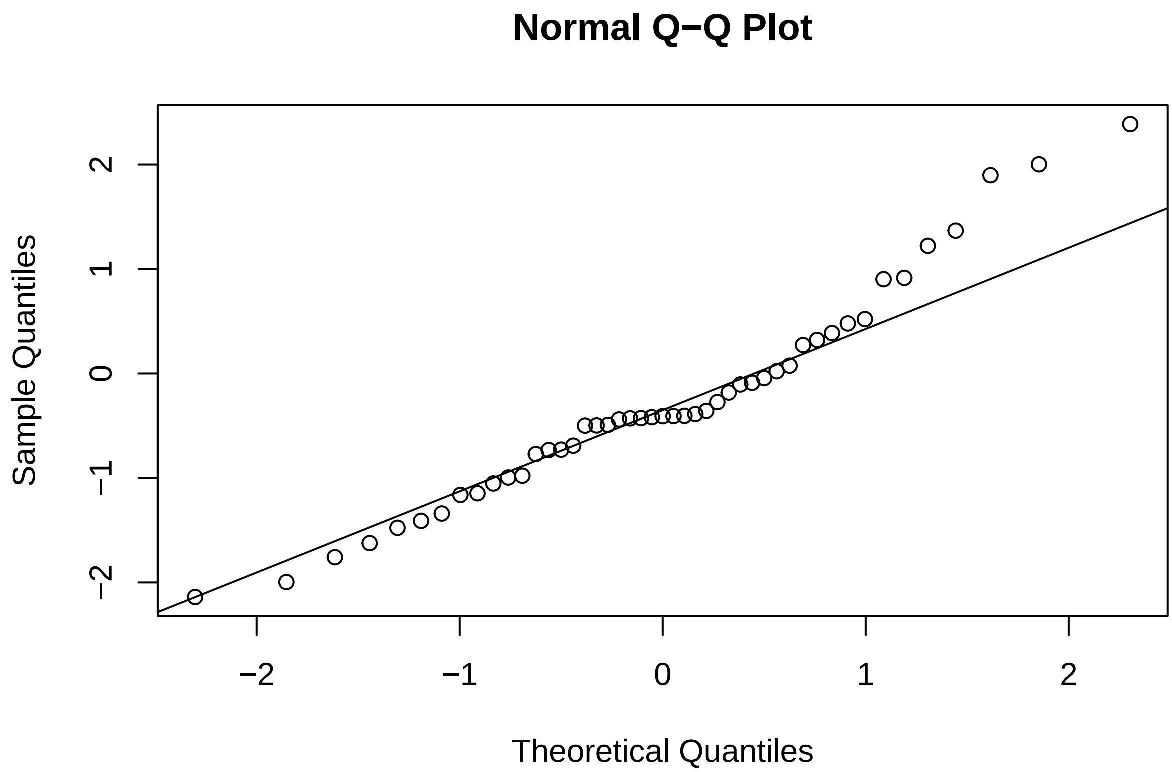

Figure 1.2: A quantile-quantile plot for the baseline model.

Developing the Logarithmic Model

```
# Performing a logarithmic transformation on the response
# variable, House_Price_Index
log.model <- stan_glm(log(House_Price_Index) ~ Date+
                       Stock_Price_Index+
                       Inflation_Rate+
                       Mortgage_Rate+
                       Unemployment_Rate+
                       Population_Growth_Rate,
                     data=data.annual,
                     refresh=0)
log.model
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      log(House_Price_Index) ~ Date + Stock_Price_Index + Inflation_Rate +
##     Mortgage_Rate + Unemployment_Rate + Population_Growth_Rate
```

```
##  observations: 47
##  predictors:   7
## ------
##                        Median MAD_SD
## (Intercept)            -105.4   11.4
## Date                      0.1    0.0
## Stock_Price_Index         0.0    0.0
## Inflation_Rate            0.0    0.0
## Mortgage_Rate             0.0    0.0
## Unemployment_Rate         0.0    0.0
## Population_Growth_Rate    0.1    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.1    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Evaluating Assumptions of the Logarithmic Model

```r
# Displaying the residual plot for this baseline model
log.fitted <- fitted(log.model)
log.resid <- resid(log.model)
ggplot(data.annual, aes(x=log.fitted,y=log.resid)) +
  geom_point() +
  geom_abline(slope=0,color='red') +
  labs(title="Logarithmic Model\nResidual Plot",
       x="Fitted HPI Values",
       y="Residuals")
```

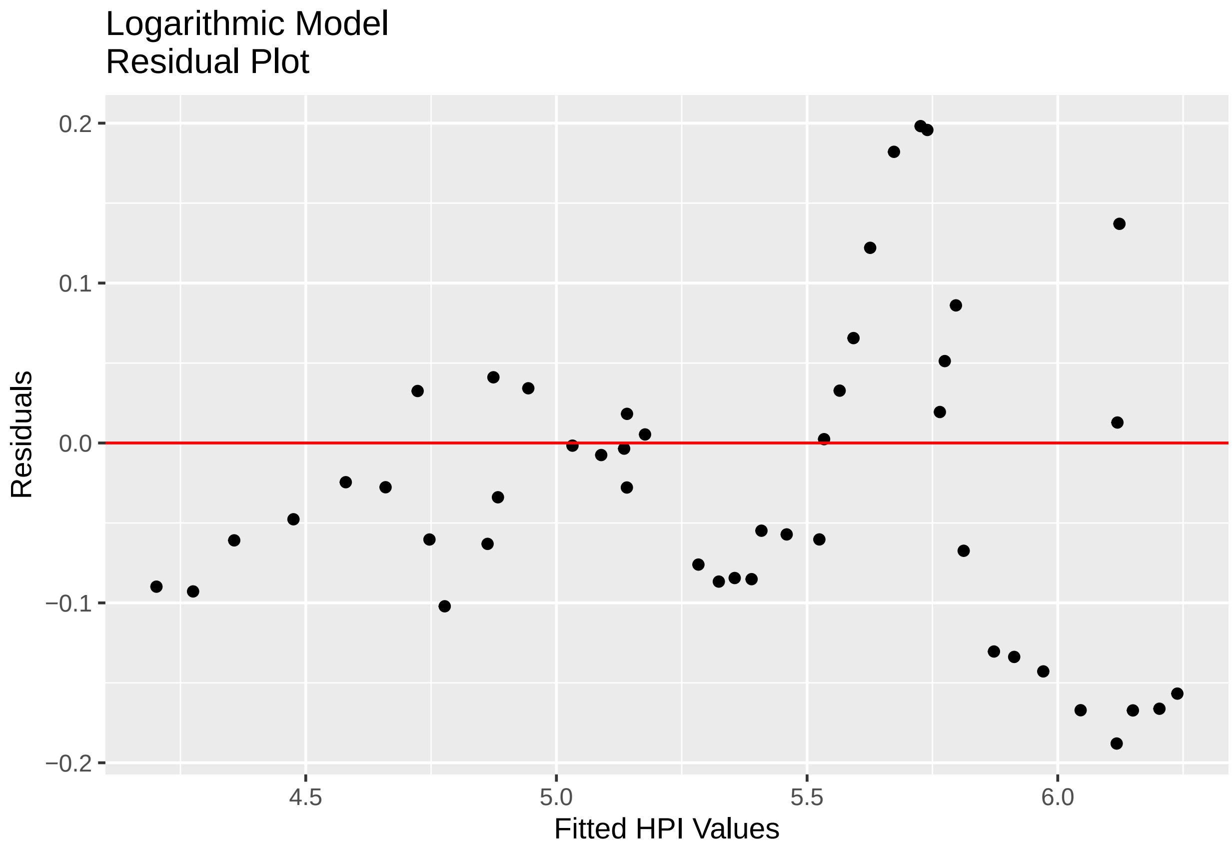

Figure 2.1: A residual vs. fitted plot for the logarithmic model.

```
log.stand.resid <- log.resid/sd(log.resid)
qqnorm(log.stand.resid)
qqline(log.stand.resid)
```

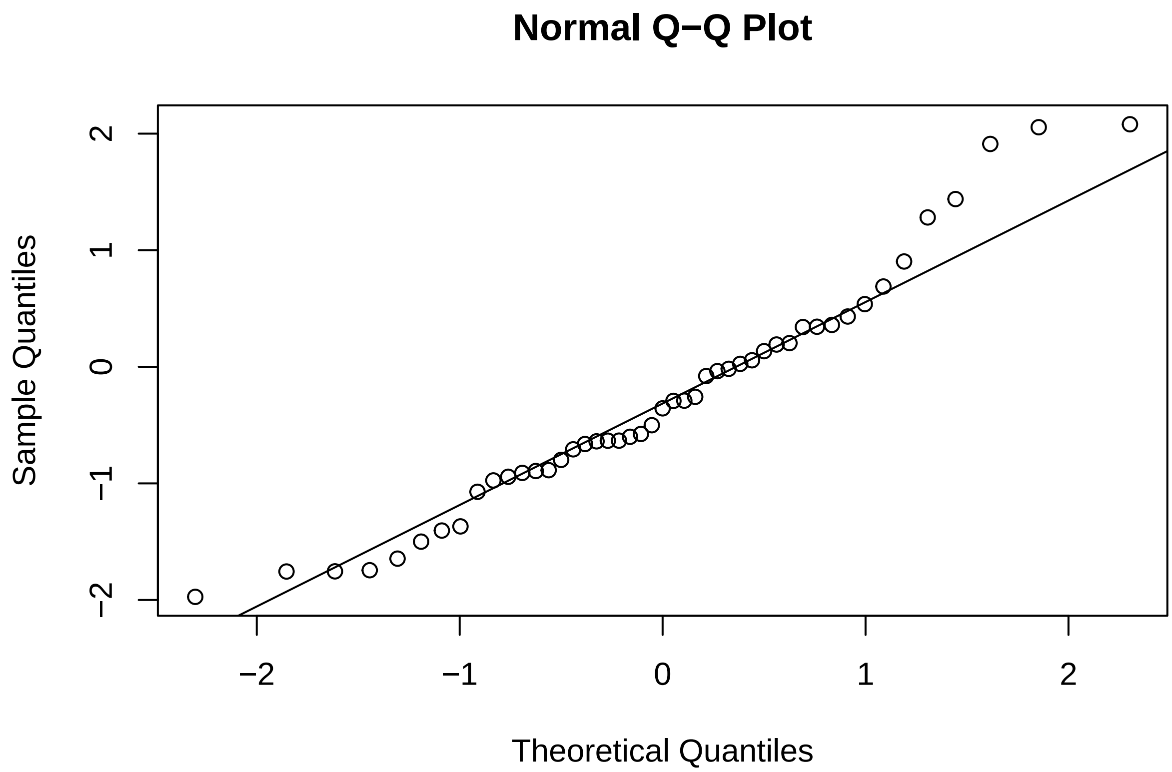

Figure 2.2: A quantile-quantile plot for the logarithmic model.

Removing Date from the Model

```r
# Including an interaction between mortgage interest rates and the Date
non_date.model <- stan_glm(log(House_Price_Index) ~ Mortgage_Rate+
                       Stock_Price_Index+
                       Inflation_Rate+
                       Unemployment_Rate+
                       Population_Growth_Rate,
                     data=data.annual,
                     refresh=0)
non_date.model
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      log(House_Price_Index) ~ Mortgage_Rate + Stock_Price_Index +
##     Inflation_Rate + Unemployment_Rate + Population_Growth_Rate
##  observations: 47
##  predictors:   6
```

```
## ------
##                         Median MAD_SD
## (Intercept)             3.8    0.4
## Mortgage_Rate           0.0    0.0
## Stock_Price_Index       0.0    0.0
## Inflation_Rate          0.0    0.0
## Unemployment_Rate       0.0    0.0
## Population_Growth_Rate 0.7    0.2
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.2    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Evaluating Assumptions of the Non-Date Model

```r
# Displaying the residual plot for the non-date model
non_date.fitted <- fitted(non_date.model)
non_date.resid <- resid(non_date.model)
ggplot(data.annual, aes(x=non_date.fitted,y=non_date.resid)) +
  geom_point() +
  geom_abline(slope=0,color='red') +
  labs(title="Non-Date Model\nResidual Plot",
       x="Fitted HPI Values",
       y="Residuals")
```

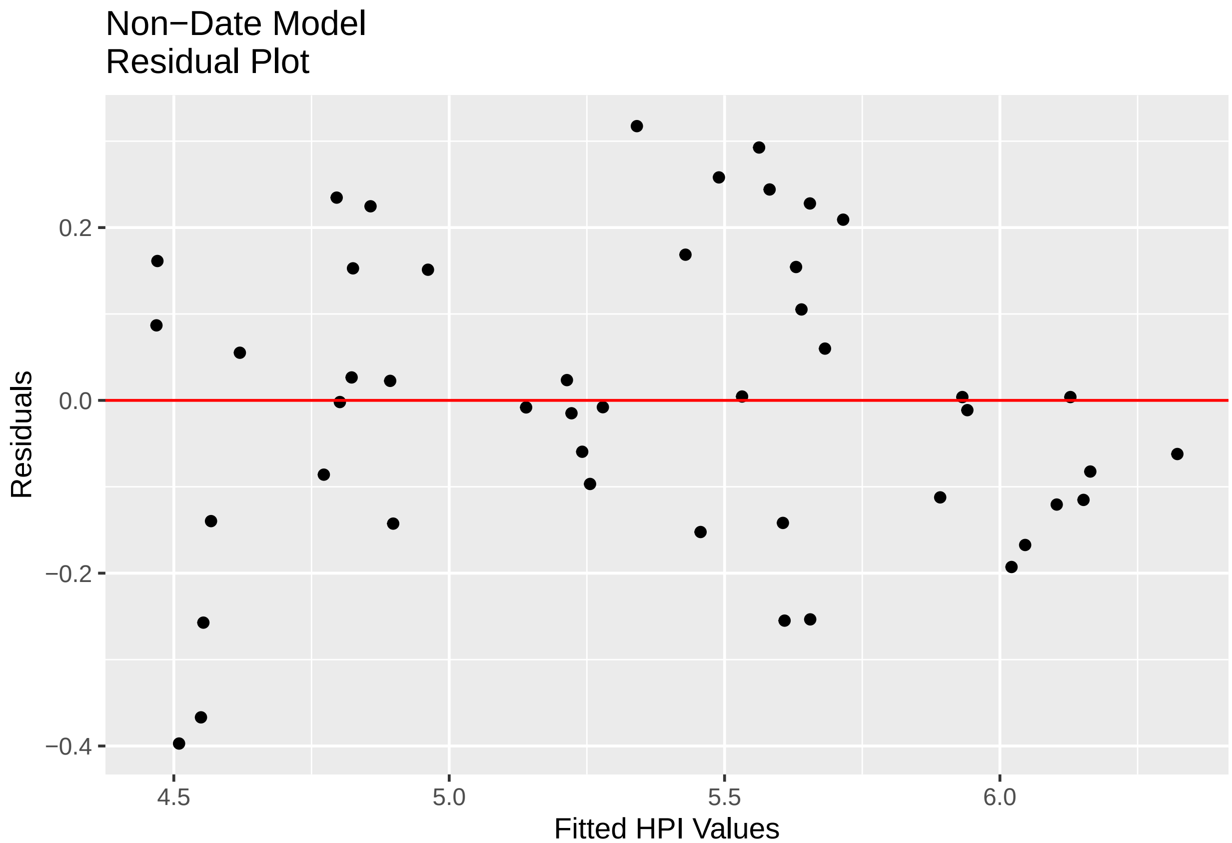

Figure 3.1: A residual vs. fitted plot for the non-date model.

```
non_date.stand.resid <- non_date.resid/sd(non_date.resid)
qqnorm(non_date.stand.resid)
qqline(non_date.stand.resid)
```

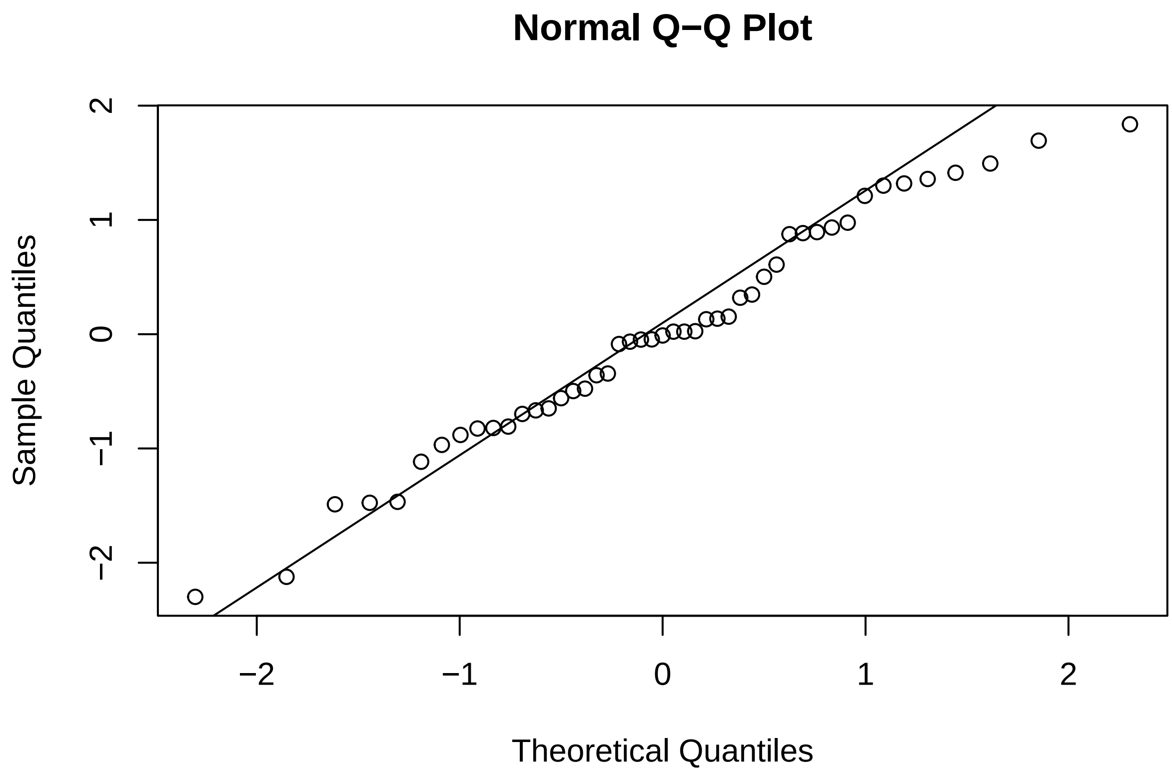

Figure 3.2: A quantile-quantile plot for the non-date model.

Evaluating the Performance of the non-Date Model

```
# Extracting the posterior predictions
checks.p1 <- posterior_predict(non_date.model)
checks.p2 <- posterior_predict(fa.model)
# Exponentiating (it's a word) non-date
# model's predictions
checks.p1 <- exp(checks.p1)

# Computing the density of each of the PPCs
checks.d1 <- density(checks.p1)
checks.d2 <- density(checks.p2)

# Creating the density plots for the posterior
# predictions
par(mfrow=c(1,2))
plot(checks.d1,col='grey',main='Non-Date Model',xlab='House Price Index (HPI)')
for(i in 2:nrow(checks.p1)){
  lines(density(checks.p1[i,]),col='grey')
```

```
}
lines(density(data.annual$House_Price_Index))


plot(checks.d2,col='grey',main='Baseline Model',xlab='House Price Index (HPI)')
for(i in 2:nrow(checks.p2)){
  lines(density(checks.p2[i,]),col='grey')
}
lines(density(data.annual$House_Price_Index))
```

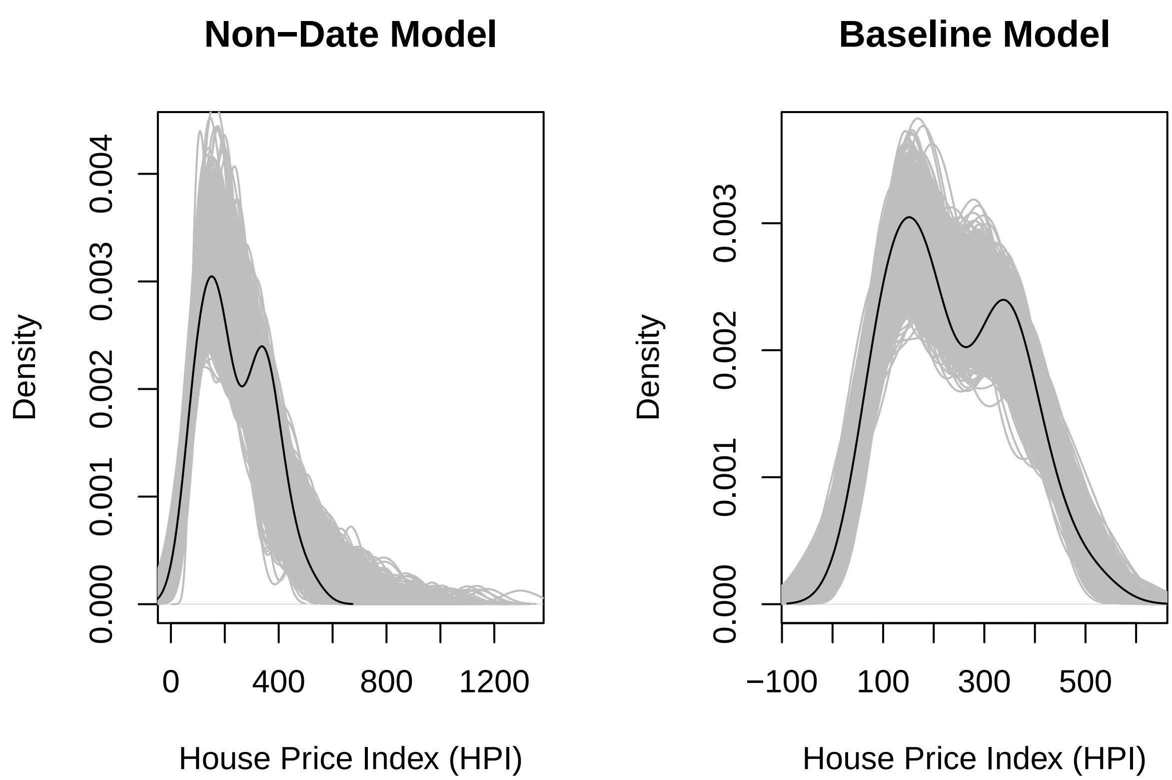

Figure 3.3: Posterior predictive check plots, where the top plot is using the non-date model and the bottom plot is using the baseline model.

```
perf.non_date <- ggplot(data.annual, aes(x=Date)) +
  geom_line(aes(y=House_Price_Index, color='Truth'),lwd=1) +
  geom_line(aes(y=exp(non_date.fitted),color='Prediction'),lwd=1) +
  labs(title="Non-Date Model",
       x="Date (year)",
       y="HPI") +
  scale_color_manual(name='Trend Line',
                     values=c('Truth'='blue','Prediction'='red'))
```

```
perf.fa <- ggplot(data.annual, aes(x=Date)) +
  geom_line(aes(y=House_Price_Index, color='Truth'),lwd=1) +
  geom_line(aes(y=fa.fitted,color='Prediction'),lwd=1) +
  labs(title="Baseline Model",
       x="Date (year)",
       y="HPI") +
  scale_color_manual(name='Trend Line',
                     values=c('Truth'='blue','Prediction'='red'))
perf.list <- list(perf.fa,perf.non_date)
grid.arrange(grobs=perf.list,nrow=2)
```

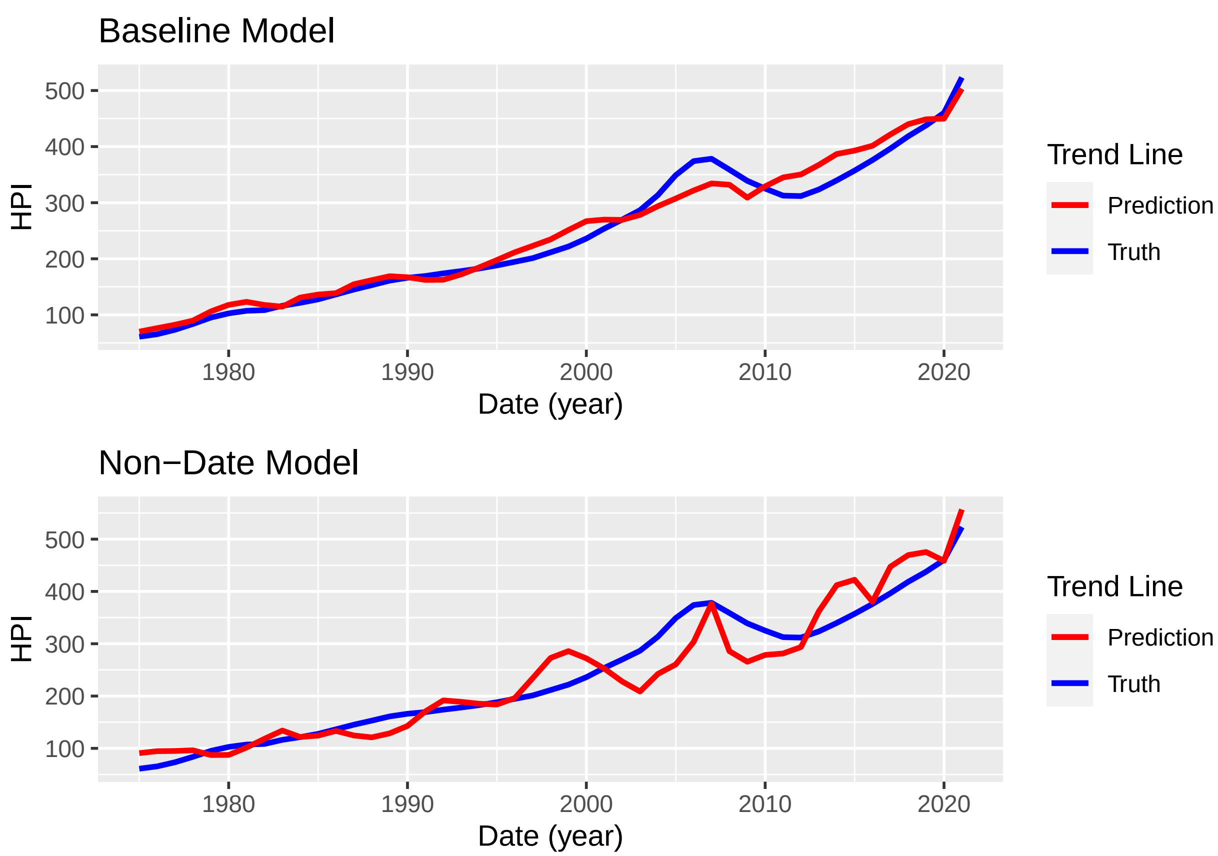

Figure 3.4: Line plots to demonstrate the accuracy of the baseline and non-date models, with the blue lines representing the "ground truth" HPI values and the red lines representing the predicted HPI values from the models.

Model Selection

Developing the Model Set

```r
# Choosing a logical set of models to test
# Second model: HPI based on Real GDP and Disposable Income
mod.set_2 <- stan_glm(log(House_Price_Index)~Real_GDP+Real_Disposable_Income,
                      data=data.annual, refresh=0)
# Third model: HPI based on Population Growth Rate and Unemployment Rate
mod.set_3 <- stan_glm(log(House_Price_Index)~
                        Population_Growth_Rate+
                        Unemployment_Rate, data=data.annual, refresh=0)
# Fourth model: HPI based on Inflation
mod.set_4 <- stan_glm(log(House_Price_Index)~Inflation_Rate,
                      data=data.annual, refresh=0)
# Fifth Model: HPI based on Mortgage Interest Rates and Stock Prices
mod.set_5 <- stan_glm(log(House_Price_Index)~Mortgage_Rate+Stock_Price_Index,
                      data=data.annual,refresh=0)

# Sixth Model: Adaptation from the non-date model to include an interaction
# between the inflation and unemployment rates
mod.set_6 <- stan_glm(log(House_Price_Index)~Inflation_Rate*Unemployment_Rate+
                        Stock_Price_Index+
                        Mortgage_Rate+
                        Population_Growth_Rate,
                        data=data.annual,refresh=0)
```

Conducting LOO Cross-Validation

```r
# Running LOO on all the models
loo.mod_1 <- loo(non_date.model)
loo.mod_2 <- loo(mod.set_2)
loo.mod_3 <- loo(mod.set_3)
loo.mod_4 <- loo(mod.set_4)
loo.mod_5 <- loo(mod.set_5)
loo.mod_6 <- loo(mod.set_6,k_threshold=0.7)
```

```
## All pareto_k estimates below user-specified threshold of 0.7.
## Returning loo object.
```

```r
# Comparing the LOO across these models, we see that Model 6 is the "best"
loo_compare(loo.mod_1,loo.mod_2,loo.mod_3,loo.mod_4,loo.mod_5, loo.mod_6)
```

```
##                  elpd_diff se_diff
## mod.set_2          0.0       0.0
```

```
## mod.set_6       -9.8       6.4
## non_date.model -14.9       4.1
## mod.set_5      -24.1       2.4
## mod.set_4      -52.4       5.6
## mod.set_3      -54.5       3.6
```

Table 1: The difference in expected log-predictive densities (ELPD) for the five models

```
# Displaying the LOO results for Model 2, we see that the true number of
# parameters in this model falls short of the p_loo estimate (2 vs. 4.5)
loo.mod_2
```

```
##
## Computed from 4000 by 47 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo     23.8  6.2
## p_loo         4.7  1.5
## looic       -47.6 12.3
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     45    95.7%   641
##  (0.5, 0.7]   (ok)        2     4.3%   228
##    (0.7, 1]   (bad)       0     0.0%   <NA>
##    (1, Inf)   (very bad)  0     0.0%   <NA>
##
## All Pareto k estimates are ok (k < 0.7).
## See help('pareto-k-diagnostic') for details.
```

Checking Assumptions for Model 2

```
# Displaying the residual plot for Model 2
mod_2.fitted <- fitted(mod.set_2)
mod_2.resid <- resid(mod.set_2)
ggplot(data.annual, aes(x=mod_2.fitted,y=mod_2.resid)) +
  geom_point() +
  geom_abline(slope=0,color='red') +
  labs(title="Model 2\nResidual Plot",
       x="Fitted HPI Values",
       y="Residuals")
```

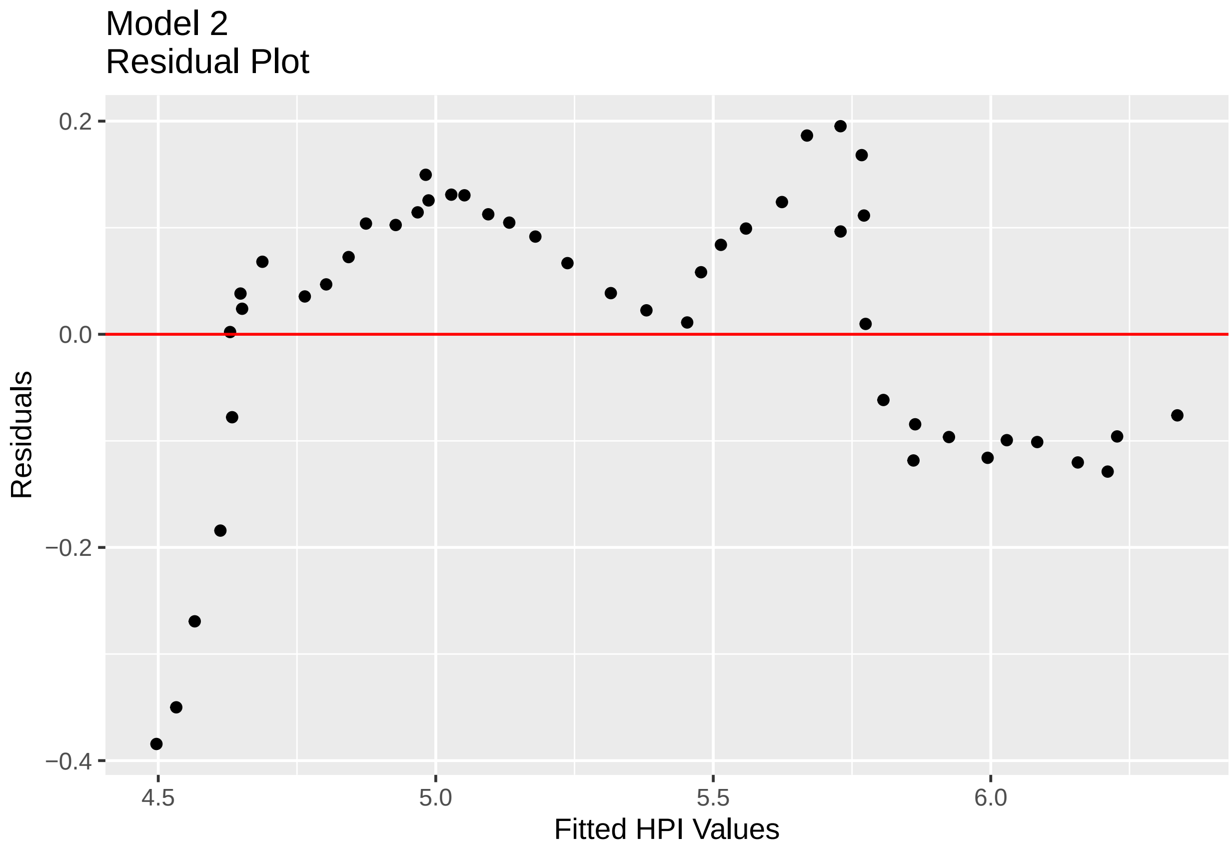

Figure 4.1: A residual plot for Model 2, as defined in the model set.

```
mod_2.stand.resid <- mod_2.resid/sd(mod_2.resid)
qqnorm(mod_2.resid)
qqline(mod_2.resid)
```

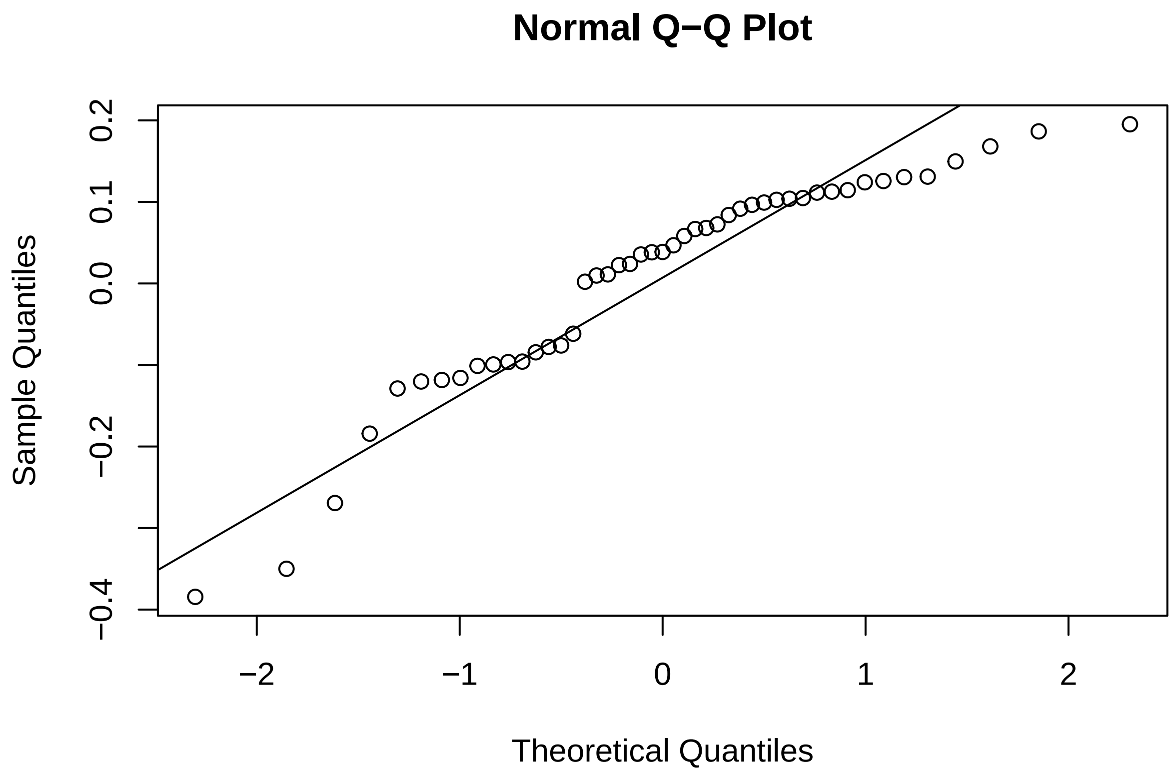

Figure 4.2: A quantile-quantile plot for Model 2, as defined in the model set.

Checking Assumptions for Model 6

```r
# Displaying the residual plot for Model 6
mod_6.fitted <- fitted(mod.set_6)
mod_6.resid <- resid(mod.set_6)
ggplot(data.annual, aes(x=mod_6.fitted,y=mod_6.resid)) +
  geom_point() +
  geom_abline(slope=0,color='red') +
  labs(title="Model 6\nResidual Plot",
       x="Fitted HPI Values",
       y="Residuals")
```

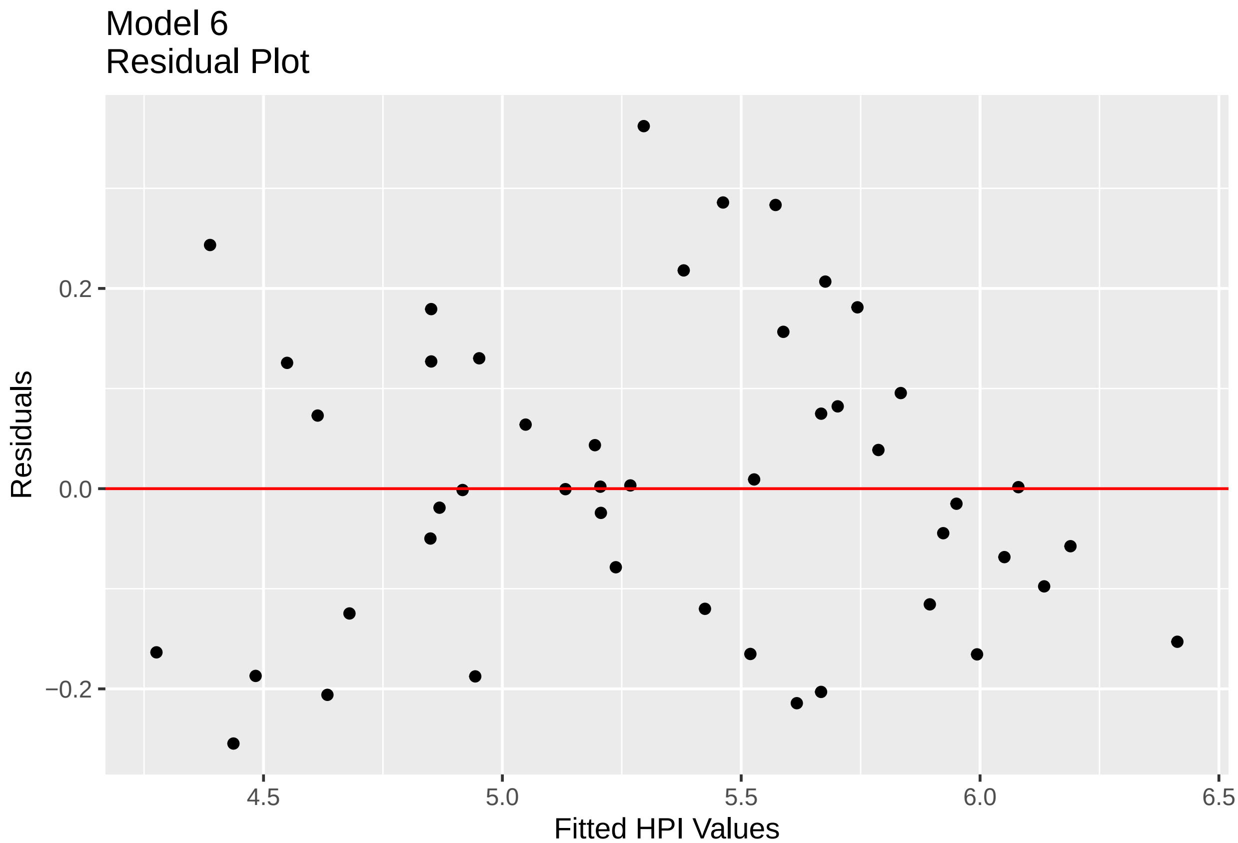

Figure 5.1: A residual plot for Model 6, as defined in the model set.

```
mod_6.stand.resid <- mod_6.resid/sd(mod_6.resid)
qqnorm(mod_6.resid)
qqline(mod_6.resid)
```

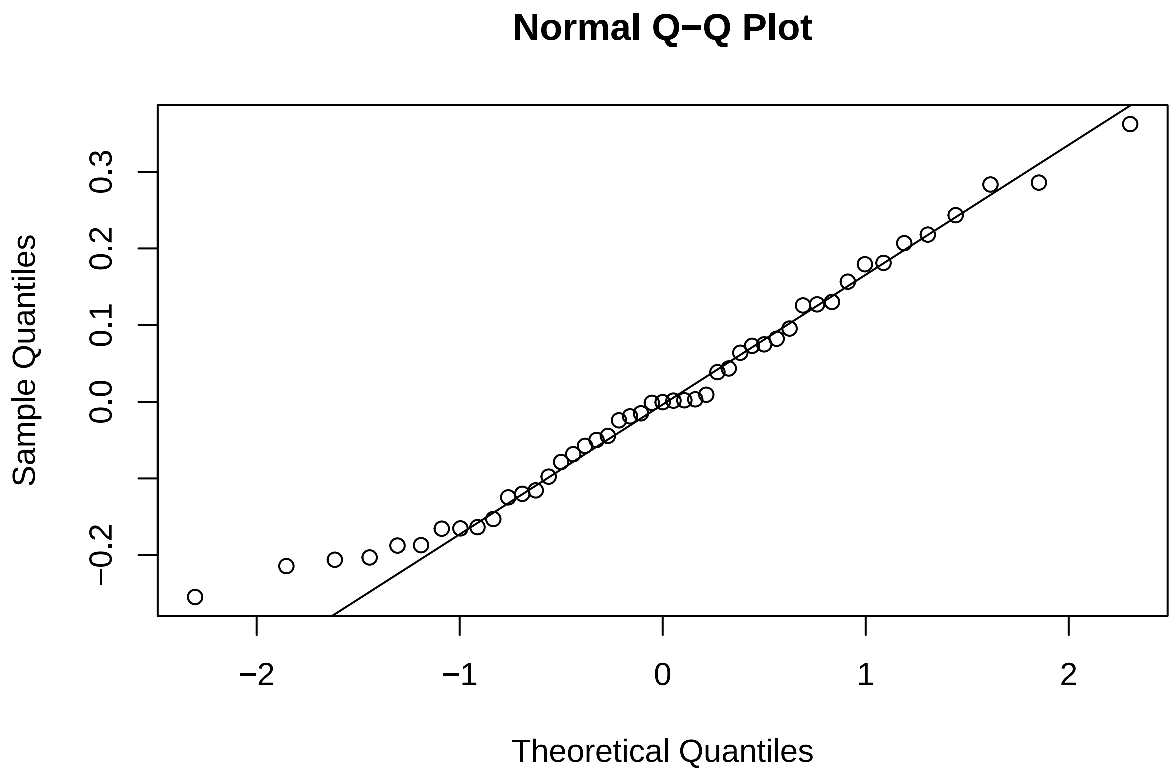

Figure 5.2: A quantile-quantile plot for Model 6, as defined in the model set.

Assessing the Best Model

```
# Displaying the LOO results for the
# "best model," we see that the number of
# effective parameters nearly match the true
# number of parameters in this model
loo.mod_6
```

```
##
## Computed from 4000 by 47 log-likelihood matrix
##
##          Estimate  SE
## elpd_loo     13.9 4.0
## p_loo         6.9 1.3
## looic       -27.9 7.9
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
```

```
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     44    93.6%   666
##  (0.5, 0.7]   (ok)        3     6.4%   387
##    (0.7, 1]   (bad)       0     0.0%   <NA>
##    (1, Inf)   (very bad)  0     0.0%   <NA>
##
## All Pareto k estimates are ok (k < 0.7).
## See help('pareto-k-diagnostic') for details.
```

```r
# Assessing model fit
mod.set_6.b_R2 <- bayes_R2(mod.set_6)
median(mod.set_6.b_R2)
```

```
## [1] 0.9168935
```

```r
# Assessing overfitting
mod.set_6.l_R2 <- loo_R2(mod.set_6)
```

```
## Warning: Some Pareto k diagnostic values are slightly high. See help('pareto-k-diagno
```

```r
median(mod.set_6.l_R2)
```

```
## [1] 0.9022436
```

```r
# Exploring the output of the model
print(mod.set_6,digits=3)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      log(House_Price_Index) ~ Inflation_Rate * Unemployment_Rate +
##     Stock_Price_Index + Mortgage_Rate + Population_Growth_Rate
##  observations: 47
##  predictors:   7
## ------
##                                 Median MAD_SD
## (Intercept)                      3.033  0.439
## Inflation_Rate                   0.146  0.056
## Unemployment_Rate                0.105  0.030
## Stock_Price_Index                0.001  0.000
## Mortgage_Rate                    0.036  0.016
## Population_Growth_Rate           0.657  0.180
```

```
## Inflation_Rate:Unemployment_Rate -0.029  0.008
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.165  0.019
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
# Plotting posterior distributions for the
# non-date model's parameters
iter <- as.matrix(mod.set_6)
mcmc_areas(iter,area_method='scaled height')
```

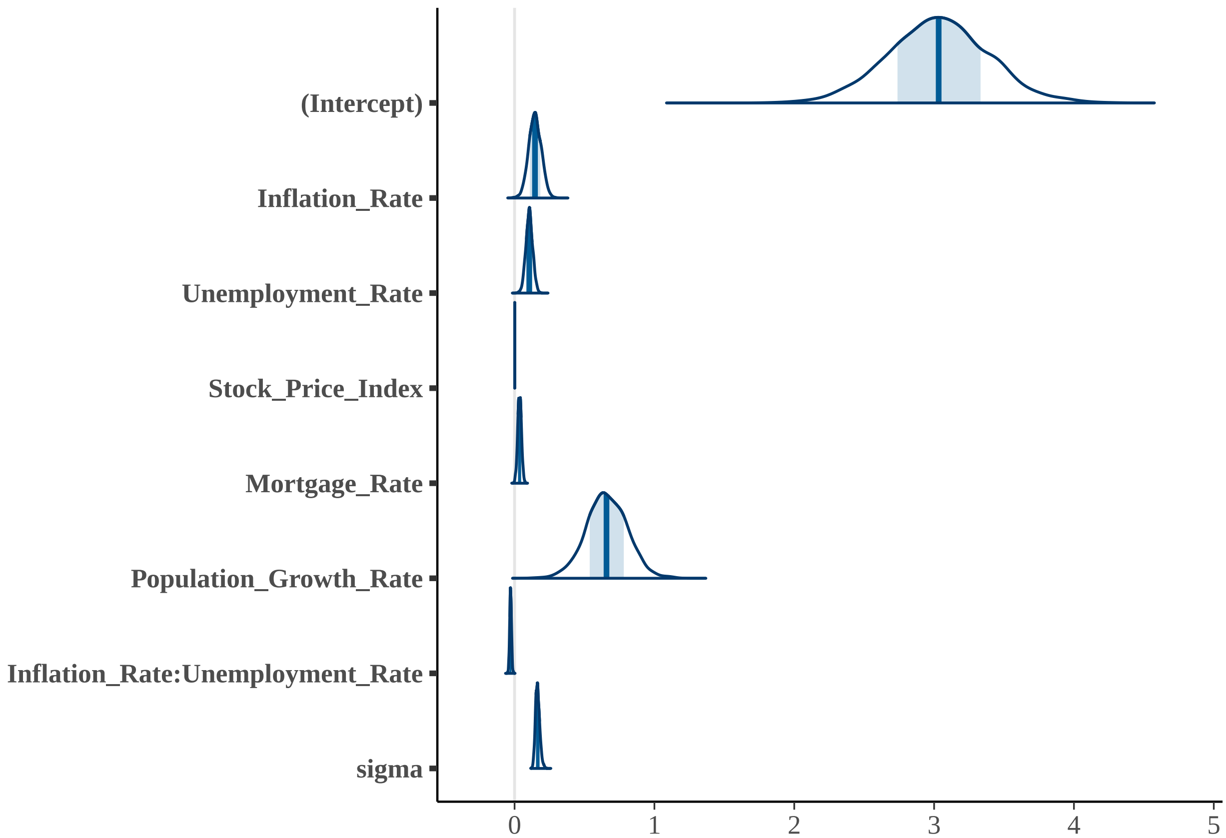

Figure 6.1: Posterior distributions for the regression coefficients of each of the explanatory variables in the non-date model, describing their relationships with the HPI.

```
# Creating an observed vs. fitted plot for Model 6
perf.mod_6.obs_fit <- ggplot(data.annual, aes(x=House_Price_Index,
                                              y=exp(mod_6.fitted))) +
  geom_point() +
```

```r
  geom_abline(color='red',slope=1) +
  labs(title='Model 6\nObserved vs. Fitted',x='True HPI',y='Predicted HPI')
# Creating an accuracy line plot for Model 6
perf.mod_6.line <- ggplot(data.annual, aes(x=Date)) +
  geom_line(aes(y=exp(mod_6.fitted),col='Prediction'),lwd=1) +
  geom_line(aes(y=House_Price_Index,col='Truth'),lwd=1) +
  labs(title='Model 6\nAccuracy Trends',x='Date (year)',y='HPI') +
  scale_color_manual(name='Trend Line',values=c('Prediction'='red',
                                                 'Truth'='blue'))
# Displaying these plots on top of each other
perf.mod_6.list <- list(perf.mod_6.obs_fit,perf.mod_6.line)
grid.arrange(grobs=perf.mod_6.list,nrow=2)
```

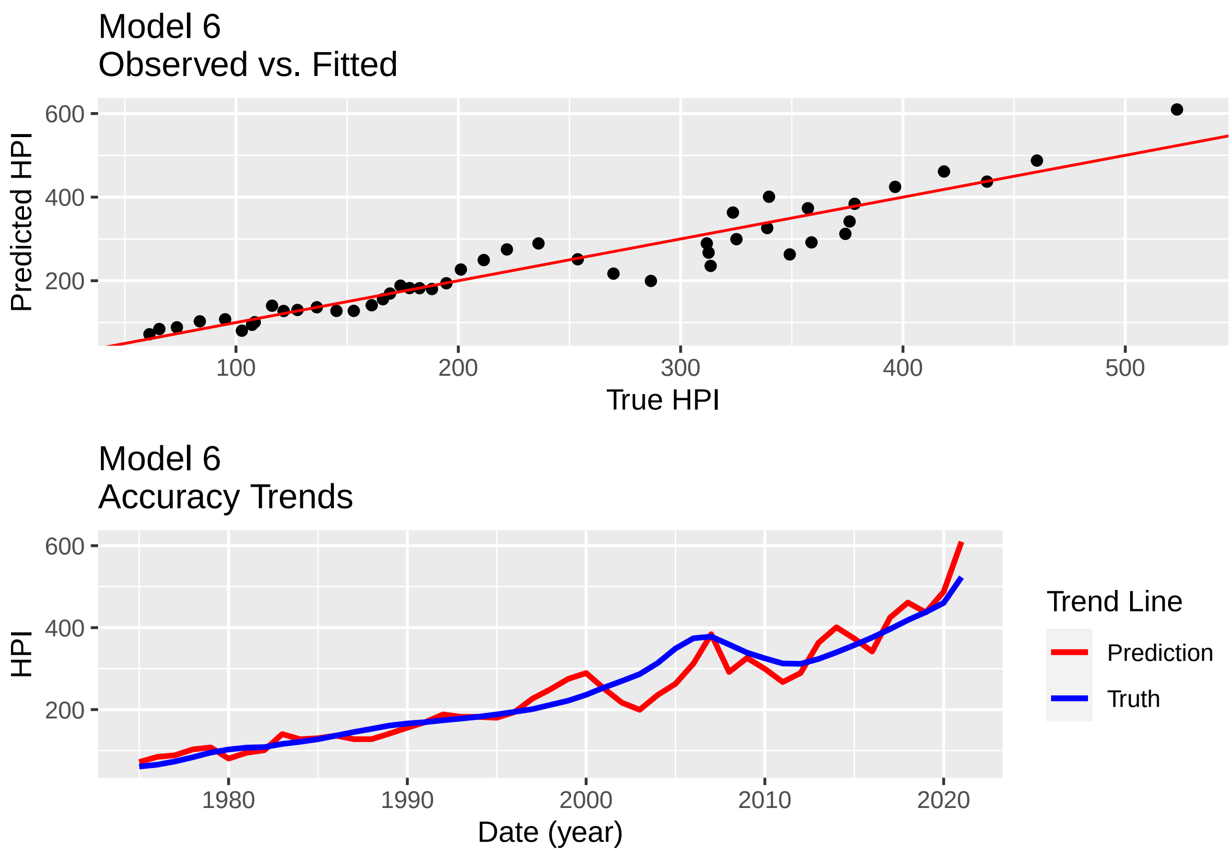

Figure 6.2: Two plots used to assess accuracy, with an observed vs. fitted plot at the top and a line plot at the bottom, where the blue line represents the true HPI values and the red line represents the predicted HPI values.

```r
# Extracting the posterior predictions for Model 6
checks.p3 <- posterior_predict(mod.set_6)
# Exponentiating (it's a word) Model 6's
# model's predictions
checks.p3 <- exp(checks.p1)
```

```r
# Computing the density of each of the PPCs
checks.d3 <- density(checks.p1)

# Plotting
plot(checks.d1,col='grey',main='Model 6',xlab='House Price Index (HPI)')
for(i in 2:nrow(checks.p1)){
  lines(density(checks.p1[i,]),col='grey')
}
lines(density(data.annual$House_Price_Index))
```

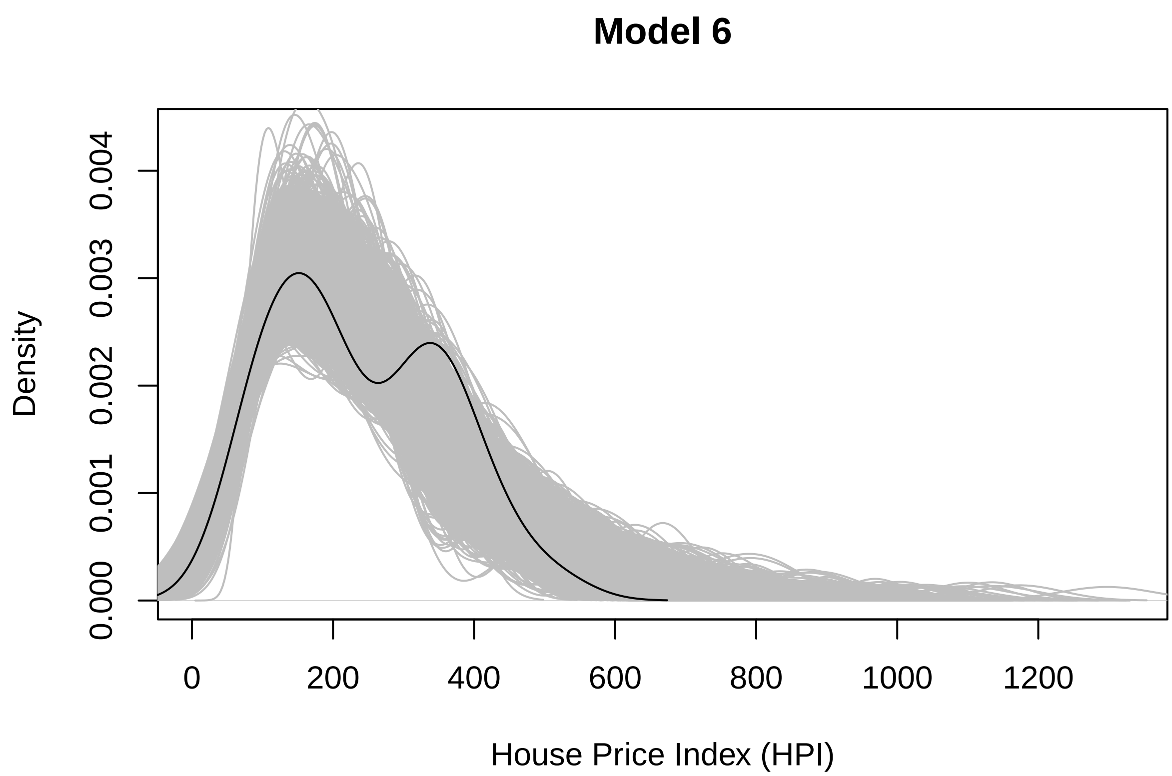

Figure 6.3: A posterior predictive check plot for Model 6, used to respectively compare simulated HPI values against the true HPI values and ultimately evaluate the predictive performance of this model.

Report

Methods

Beginning with a FRED data set containing the initial variables in the data set, note that we created the variable `Inflation_Rate` to model the percentage change in consumer price index, thus ensuring that each observation, that is, each year, is independent from one another. Furthermore, for simplicity, we reformatted the date variable to only include the year as opposed to the format `01-01-YEAR`. Since one of the research questions involved examining key recessions throughout the time period 1975-2021, we have opted to not remove any "outliers" from the data set, such as the year 2021, which was a part of The Great Resignation and reported a House Price Index (HPI) of 523.26.

When constructing each model to predict the response variable, `House_Price_Index`, we used Bayesian multiple linear regression to determine the regression parameters that best fit the data given the explanatory variables. In the initial model, which we will refer to as the baseline model, we included `Date`, `Stock_Price_Index`, `Inflation_Rate`, `Mortgage_Rate`, `Unemployment Rate`, and `Population_Growth_Rate` as explanatory variables. In addition, we constructed a secondary model, which we will refer to as the non-date model, that mimicked the baseline model but included a logarithmic transformation of the response variable and excluded the date variable. As shown in various figures above, we analyzed residual plots and quantile-quantile plots when assessing the assumptions of linearity, constant variance, and normality.

Proceeding forward, we defined a model set including six distinct models, all of which having a logarithmic transformation of the response variable, `House_Price_Index`. Note that the first of these models is the non-date model, as described previously. The second model included `Real_GDP` and `Real_Disposable_Income` as explanatory variables because of the fact that having more disposable income makes individuals more willing to invest in housing, increasing GDP and thus raising housing prices in the short run. The third model included

`Population_Growth_Rate` and `Unemployment_Rate` as explanatory variables, as when the population size increases, so does the demand for occupations, decreasing the unemployment rate in the short run and thus increasing the demand for housing, ultimately causing housing prices to rise. The fourth model included only `Inflation_Rate` as an explanatory variable because clearly, when the inflation rate increases, so does the price level, thus increasing housing prices. The fifth model included `Mortgage_Rate` and `Stock_Price_Index` as explanatory variables since higher mortgage rates make borrowing more expensive, decreasing the demand for housing and thus lowering housing prices, and on the contrary, higher stock prices cause shareholders to gain more value in their portfolios, making them more inclined to invest in housing and thus raising housing prices. Finally, the sixth model mimics the first model with a slight modification, that being the inclusion of an interaction between `Inflation_Rate` and `Unemployment_Rate`, as the Phillips Curve models an inverse relationship between these two variables, implying that they may have an interaction effect on housing prices. To compare all these models, we used leave-one-out (LOO) cross-validation to both calculate the expected log predictive densities (ELPD) and assess over-fitting.

Results

Note that the baseline model serves as the largest reasonable model that aligns with our research questions, which seek to address which macroeconomic variables have a strong influence on housing prices and whether severe economic recession significantly factor into housing prices. Beginning with this model, we saw clear violations of the assumptions of linearity and constant variance in Figure 1.1, as the spread of the residuals is not consistent throughout the plot and there is also some curvature in the region with the fitted HPI values above 200. Figure 1.2 also demonstrated that the residuals are slightly skewed but roughly normally distributed. In an attempt to resolve these violations, we first performed a logarithmic transformation on the response variable, but as shown in Figures 2.1 and 2.2, this did not change the results significantly. Ultimately, it became clear that the problem with these models was our inclusion of `Date` as an explanatory variable, as `Date` and `House_Price_Index` most likely have a non-linear relationship since the U.S. economy traverses through the different stages of the business cycle over the course of time, those respectively being the peak, recession, trough, and expansion. Hence, we proceeded to create the non-date model, as described previously, and Figures 3.1 and 3.2 show clear resolutions of all the previous violations, as the residuals are randomly scattered and still approximately normally distributed.

Evaluating the performance of the non-date model, we concluded based on Figure 3.3 that the predictive performance of the baseline model is stronger than that of the non-date model, as with the baseline model, the simulated predictions better match the true HPI for each year. Note, however, that in the posterior predictive plot for the baseline model, there are negative predicted HPI values, contradicting the definition of the House Price Index, reinforcing our decision to apply a logarithmic transformation to the response variable, as this constrains the value of HPI to be positive. In Figure 3.4, there are lots of instances where the non-date model either underestimates or overestimates the HPI for a respective year, while the baseline model underestimates the HPI during the 2008 Great Recession.

Performing LOO cross-validation on the model set, as defined previously, we saw that the second model, which included `Real_GDP` and `Real_Disposable_Income` as explanatory variables, had the lowest ELPD and thus the lowest LOO information criterion (LOOIC). Checking the assumptions for this model, however, it failed to satisfy constant variance, linearity, and normality, as evident by Figures 4.1 and 4.2. Hence, we proceeded to evaluate the next best model as determined by Table 1, that being the sixth model, which modified the non-date model to include an interaction between `Inflation_Rate` and `Unemployment_Rate`. Based on Figures 5.1 and 5.2, it is clear that this model meets the assumptions of linearity, constant variance, and normality, allowing us to move on to assessing its predictive performance.

Note that Model 6 reported a Bayesian R^2 value of 0.917, indicating that the model explained a very high percentage of variability in the data. The LOO R^2 value for this model was 0.901, only slightly undermining the Bayesian R^2 value, implying that over-fitting is not a concern for this model, despite its six parameters. Examining Figure 6.2, it is evident that this model has a high accuracy of predicting the true HPI in a given year, as the predicted HPI values closely match their respective fitted HPI values. Figure 6.3's posterior predictive check plot demonstrates, however, that a lot of the simulated HPI values do not quite match the true HPI values, particularly in the right side of the plot, indicating that this model's predictive performance outside the data may not be as strong as suggested by Figure 6.2. Finally, analyzing the posterior distributions of the model coefficients in Figure 6.1, it is clear that a majority of the regression coefficients excluded zero, suggesting a relationship between nearly all the explanatory variables and the response variable. For example, the effects of `Population_Growth_Rate`, `Unemployment_Rate`, `Inflation_Rate`, and `Mortgage_Rate` were generally positive compared to the baseline, with the exception of the interaction effect between `Inflation_Rate` and `Unemployment_Rate`, which has a regression coefficient of -0.028 and a 95% uncertainty interval of $[-0.044, -0.012]$. Note that it is challenging to determine visually whether `House_Price_Index` increases with `Stock_Price_Index`, but the 95% uncertainty interval and regression coefficient of 0.001 suggests that it does.

Discussion

This study sought to determine which macroeconomic variables have a significant effect on housing prices to help individuals determine the ideal time to invest in housing based on the current state of the economy. We also noted that certain recessions can impact these variables, thus influencing housing prices, as seen with The Great Recession (2007-2010) in Figure 3.4. Thus, the goal of this study was to develop a model that can accurately estimate housing prices despite shocks to the economy caused by these recessions. As evident by Figure 6.2, our "sixth" model, which built upon the non-date model, was the most successful in delivering these accurate estimates.

Based on the results of this model, there is sufficient evidence to suggest that as each of our macroeconomic variables increase, so do housing prices. Note that while mortgage rates and stock prices have positive relationships with housing prices, that is, when at least one of these variables increases, so do housing prices, the results ultimately show that this relationship is weak, which implies that these variables may not be strong indicators of whether it is ideal to invest in housing in a given year. In a practical context, this implies that it may be best to invest in housing when the inflation, unemployment, and population growth rates are relatively low, avoiding giving much weight to the other macroeconomic variables in the model. Note that it is particularly challenging for there to be a period when inflation and unemployment rates are both low, as the Phillips Curve suggests that these variables have an inverse relationship; that is, when one of them increases, the other decreases. To handle this, we suggest with decision-making to give more weight to the inflation rate, as it has a stronger individual effect on housing prices than the unemployment rate. One thing to consider in particular is whether economists predict the inflation rate to increase or decrease in the future, as this clearly factors into present decision-making.

Note that the consideration of potential interactions between inflation rate and unemployment rate ultimately yielded a model with higher accuracy. Thus, one drawback of this study is that we did not consider any additional interactions between any of the other variables, such as one between the population growth rate and the unemployment rate. Hence, there may exist a model with even better performance that accounts for these interactions, but it is important to avoid "model dredging," that is, fitting every single possible model to the data and choosing the one that yields the highest accuracy. A secondary drawback of this model is that we saw based on Figure 6.3 that it may struggle to predict housing prices outside the data, making it challenging to use this model for extrapolation, which is particularly important as the U.S. economy can be unpredictable. Ultimately, however, the value of this study comes from our conclusions about which macroeconomic variables are the best indicators of housing prices, allowing individuals to monitor these variables to determine when to invest in housing.