# Mortgages

Nathan Cantafio

November 02, 2024

# Contents

# Motivations

I was playing around with calculating the total cost of a mortgage, and I realized that for $700000 home, you could easy pay $1.1 million on a mortgage. Growing up my dad has always thought that renting is a waste of money, but this got me wondering if that's actually true. Could renting sometimes be a better financial choice than buying?

```
# returns monthly payment to pay off mortgage in the specified time (in years)
# interest rate is annualized and downpayment is given as a percentage
# assumes monthly interest accrual and payments
payment <- function(principal=500000, rate=0.05, downpayment=0.05, time = 25) {
  return((rate/12)*((1+rate/12)^(12*time))*
           (principal-downpayment*principal)/((1+rate/12)^(12*time)-1))
}
300*payment(principal = 700000) # total cost of mortgage is over 1.1 million!!!
```

```
## [1] 1166257
```

This question essentially boils down to finding a way to estimate the total cost of a mortgage. For fixed rate mortgages this comes down to some linear algebra. For variable rate mortgages, we need a way to predict (or at least generate plausible) interest rates. This is a challenging problem. In this project I will develop a few different methods for doing this, and go over their pros and cons. In the end we will compare the total cost of a mortgage to a rental and see if we can say anything meaningful.

# Background

In this section we will develop some of the theory that we later apply to the problem of predicting (or generating) interest rates. This involves topics ranging from numerical methods to statistical inference/probability theory.

## Monthly payment formula for a fixed rate mortgage

To find a formula for what the monthly payment must be in order to pay off a fixed rate mortgage in a specified number of years I used linear algebra to solve the recurrence. This is unique compared to the other approaches I've seen online which used sums.

Assume that interest is accrued over the same period as payments are made. Let $a_n$ be the amount owed after $n$ payment periods. Then

$$a_n = (1+r)a_{n-1} - c,$$

where $r$ is the interest rate and $c$ is the monthly payment.

This relation can be rewritten as:

$$\begin{bmatrix} a_n \\ c \end{bmatrix} = \begin{pmatrix} 1+r & -1 \\ 0 & 1 \end{pmatrix} \begin{bmatrix} a_{n-1} \\ c \end{bmatrix},$$

and then further rewritten as:

$$\begin{bmatrix} a_n \\ c \end{bmatrix} = \begin{pmatrix} 1+r & -1 \\ 0 & 1 \end{pmatrix}^n \begin{bmatrix} a_0 \\ c \end{bmatrix} = A^n \begin{bmatrix} a_0 \\ c \end{bmatrix}.$$

Call the matrix $A$. Then $A = MDM^{-1}$ where $D$ is a diagonal matrix. In fact,

$$A^n = \begin{pmatrix} 1/r & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1+r \end{pmatrix}^n \begin{pmatrix} 0 & 1 \\ 1 & -1/r \end{pmatrix} = \begin{pmatrix} (1+r)^n & \frac{1}{r}(1-(1+r)^n) \\ 0 & 1 \end{pmatrix}.$$

And we have:

$$a_n = a_0(1+r)^n + c\frac{1-(1+r)^n}{r}.$$

If the goal is to pay off the mortgage in $N$ periods, then setting $a_N = 0$ yields:

$$c = a_0\frac{r(1+r)^N}{(1+r)^N - 1}, \qquad \text{where } a_0 \text{ is the principal.}$$

## Vasicek and CIR models

The Vasicek and CIR models use Stochastic Differential Equations to model interest rates.

The basic idea of both is that there is a value (which we call $\theta$) to which the interest rate should gravitate towards over long enough time scales.

Let $r$ be the interest rate we are modeling. Then the change in the rate, $dr$ should be in the direction $\theta - r$ so that

$$r + dr = r + \theta - r = \theta,$$

in other words $r$ is being pulled towards $\theta$.

We don't want it to be pulled all the way though, otherwise the model wouldn't be very interesting. So we add another parameter $\alpha$ which acts as the speed. Thus

$$dr = \alpha(\theta - r).$$

The above on its own is called the drift term. There is also some volatility or random noise introduced with a parameter sigma and voila:

$$dr = \alpha(\theta - r)dt + \sigma dW.$$

Above is the Vasicek model, we add a $dt$ into the equation to represent the time-step.

The CIR model is similar but volatility is also proportional to the square root of the rate:

$$dr = \alpha(\theta - r)dt + \sigma\sqrt{r}dW.$$

The CIR model is nice because at least analytically, as long as $2\alpha\theta > \sigma^2$ (Feller condition) then the rate will never be negative. This is because when $r$ becomes close to zero, the $\sqrt{r}$ term becomes very small and allows the drift term to dominate over the volatility term; pulling $r$ up. However, this does not work out so nicely in simulation due to errors introduced by discretization which can result in negative rates.

We can discretize the CIR model as follows (Forward Euler scheme):

$$r[t] = r[t-1] + \alpha(\theta - r[t-1])dt + \sigma\sqrt{r[t-1]}dW.$$

We can then rearrange this as:

$$\frac{r[t]}{\sqrt{r[t-1]}} = a\frac{r[t-1]}{\sqrt{r[t-1]}} + b\frac{1}{\sqrt{r[t-1]}} + \varepsilon,$$

where $a = 1 - \alpha dt$, $b = \alpha\theta dt$ and $\varepsilon = \sigma dW$ so that we can run a linear regression.

## Predicting (or generating) interest rates

Here is a plot of the average 30 year mortgage rate in the US from 1971 to 2024 (data from (Mac 2016))

```
# load data
usrates <- read.csv(file = "MORTGAGE30US.csv")
colnames(usrates) <- c("date", "rate")
usrates$date <- base::as.Date(usrates$date, format = "%Y-%m-%d")

# fix data so it is evenly spaced
n = length(usrates$date)
for (i in 2:n) {
  usrates$date[i] = usrates$date[i - 1] + 7
}

usratesPlot <- ggplot(data = usrates) +
  geom_line(aes(x = date, y = rate)) +
  labs(title = "Average 30 year mortgage rate", y = "Rate", x = "Date") +
  theme_bw()
usratesPlot
```

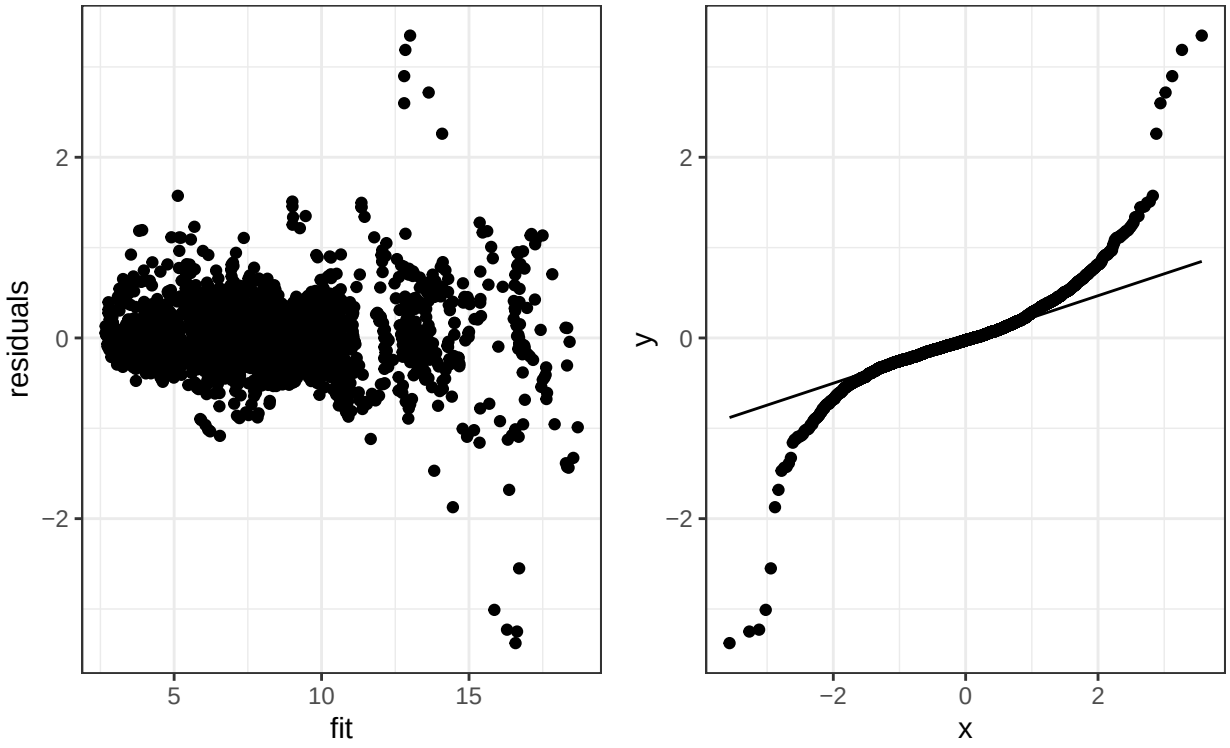

This is the data we will use to generate our predictions.

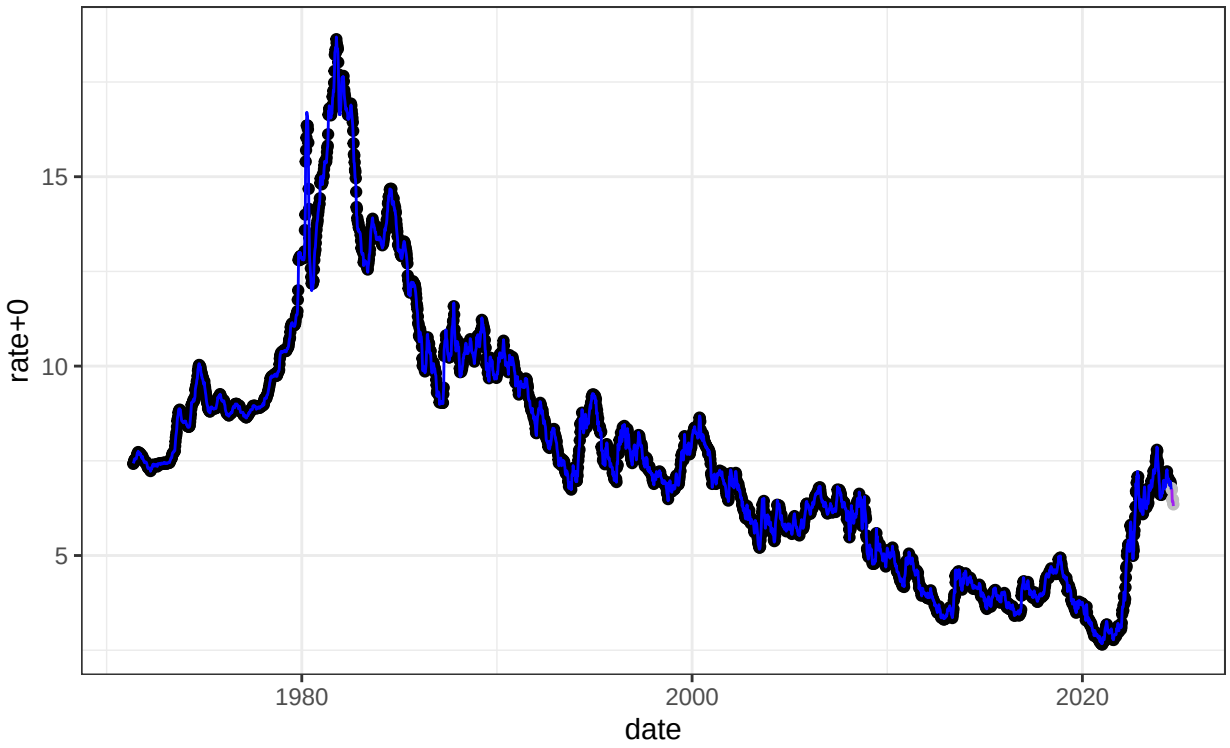
## Approach 1 - Predicting interest rates with a linear model

We will fit a linear model using OLS. We will try predicting the overnight rate $t$ months from now, using the rate now, and the rates every 6 months ago until we hit $t$ months ago. For example, we might predict the rate 18 months from now using the rates now, 6 months ago, 12 months ago, and 18 months ago.

Here is the residual plot, qq plot, and fitted model for $t = 6$. The black dots are the data that the model was trained on, the gray dots are the data that model did not get to see in training. The blue lines are the fitted values, and the purple line is future predictions.

```
rate = usrates$rate
date = usrates$date
# want to predict t months into future (t should be divisible by 6)
# coerce data into desired format
t <- 6
n = length(rate)
date = date[(t+1):(n-t)]; `rate+0` = rate[(t+1):(n-t)]; `rate+t` = rate[(2*t+1):n]
rates <- cbind.data.frame(date, `rate+t`, `rate+0`)
# rate-t, ... rate-(t-6), ... rate-6
for (i in 1:(t/6)) {
  rates <- cbind.data.frame(rates, rate[(1 + 6*(i-1)):(n - 2*t+6*(i-1))])
}
colnames(rates) <- c('date', 'rate+t', 'rate+0', paste("rate-", seq(t, 6, by=-6), sep=""))
n <- length(rates$date)
rates_test <- rates[(n-t+1):n,]
rates <- rates[(1:(n-t)),]

# regress `rate+t` on `rate+0`, `rate-6`, ..., `rate -(t-6), and `rate-t`
model <- lm(`rate+t` ~. - date, data = rates)
```

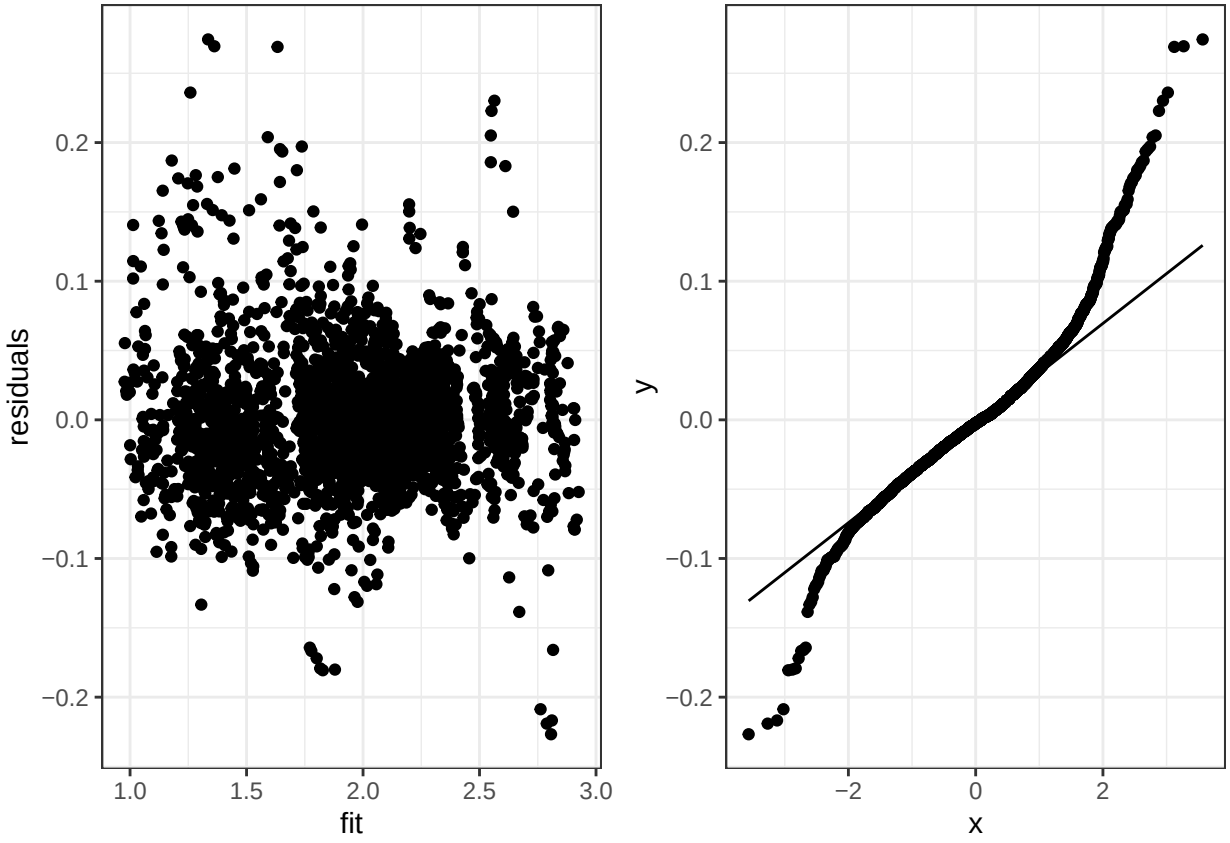





From the residuals and qq plots, the assumptions of a linear model do not seem to be met, and using a linear model does not seem justified. However, in all honesty the predictions are pretty good.

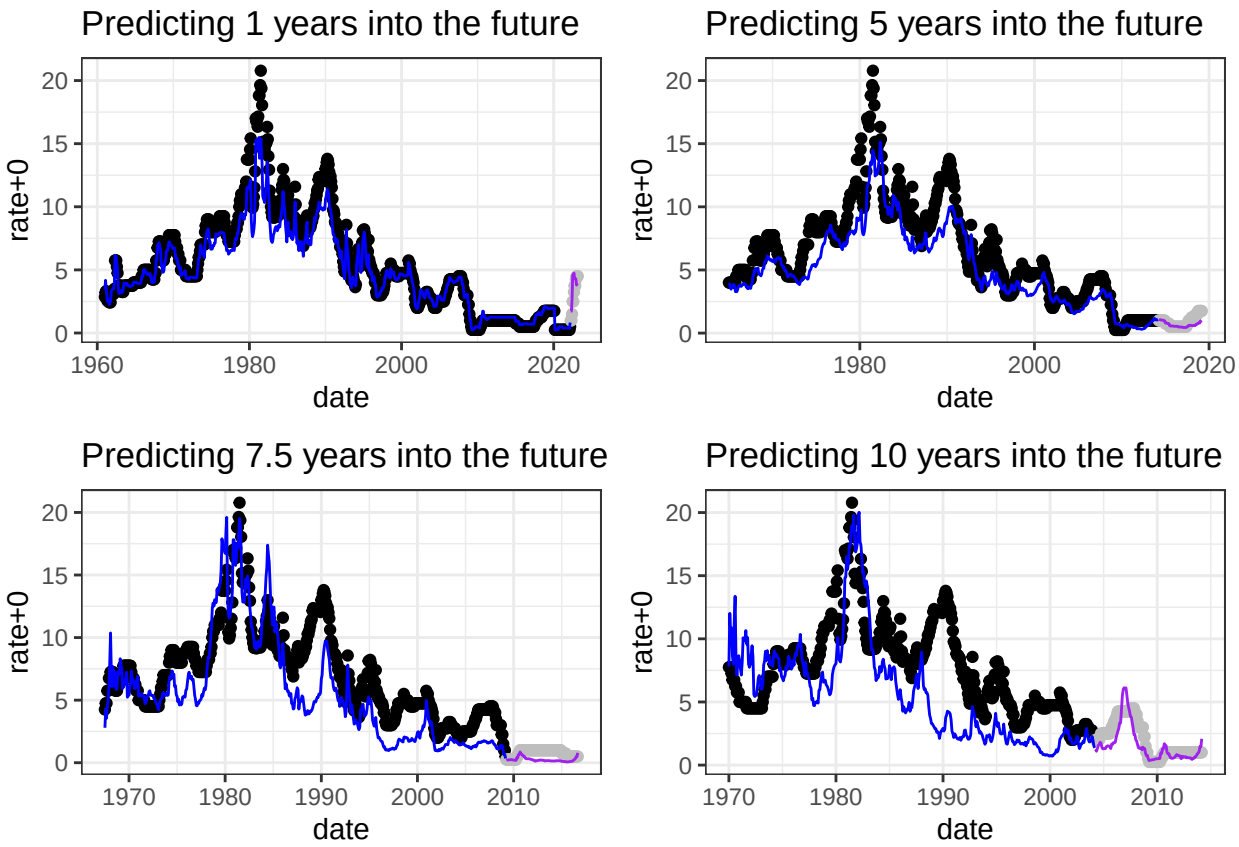
But maybe if we transform our data, we can do an even better job. Here we take the log of the rate to "mellow it out".

```
rates_tr <- rates %>%
  mutate(across(`rate+t`:`rate-6`, log))

model <- lm(`rate+t` ~. -date, data = rates_tr)
residuals <- data.frame(fit = model$fitted.values, residuals = model$residuals)
resplot <- ggplot(data = residuals) +
  geom_point(aes(x = fit, y = residuals)) +
  theme_bw()
qqplot<- ggplot(residuals, aes(sample = residuals)) +
  stat_qq() +
  stat_qq_line() +
  theme_bw()
cowplot::plot_grid(resplot, qqplot)
```




From the residual and qqplots, it seems like this made our data a little bit more suited towards a linear model. I don't mind transforming the data in this way because I am not trying to use this model for any interpretations - just making predictions.


We can see that the model does a pretty nice job of fitting the data! I would say slightly better than the untransformed version. And we even get some predictions for future interest rates in purple. Unfortunately, the further out we predict the worse our fit is.




This is unsurprising since to predict further, we need to essentially shrink our data size.

Overall I'm not a huge fan of this approach. It makes sense that subsequent interest rates would be correlated, so because of the way we set up the model, we shouldn't really make the assumption of independent predictors (which we implicitly have to by using OLS). And the predictions just aren't very useful if we go far out.

## Approach 2 - Generating plausible interest rates

Rather than trying to make predictions about future interest rates from previous rates, what if we had access to the distribution of interest rate trajectories? Then we could check how often in this distribution is, for example, the total cost of a mortgage more than double the principal?

In some sense we are trying to estimate the distribution of a vector in $\mathbb{R}^T$ where $T$ is the number of rates we have observed, and each component is likely highly correlated with each other. And all from a single observation! When I write it down like this it seems fruitless... When I first started this project I didn't have as clear an understanding of what I was doing. In any case here are the results of this first approach.

We can use SDEs to model the process.

I chose to use a CIR model to avoid negative interest rates. However, due to errors introduced by the discretization, I added a step of taking the absolute value. The function for simulating a CIR model is below.

```
cir <- function(alpha, theta, sigma, steps, start = 0) {
  dt <- 1 # time step

  r <- vector(length=steps)
  r[1] <- start # initial interest rate
  # r[1] <- 5 # could choose any initial rate we want


  for (i in 2:steps) {
    dW <- rnorm(1, mean=0, sd=1) # generate random noise

    r[i] <- abs(r[i-1] + alpha*(theta - r[i-1])*dt +      # modified euler
                  sigma*sqrt(r[i-1])*dt*dW)
  }

  return(r)
}
```

To choose parameters that yield realistic interest rates, I used data of the average mortgage rate for 30 year mortgage in the US rate from April 1971 to October 2024. I then ran a linear regression, predicting the last $n-1$ rates from the first $n-1$ rates. Using the formula from before to get the CIR parameters from the OLS coefficients. These parameters act as an initial guess for the MLE estimation which follows. I could have just stopped at the regression, but I wanted to use MLE to refine and hopefully improve the choice of parameters.

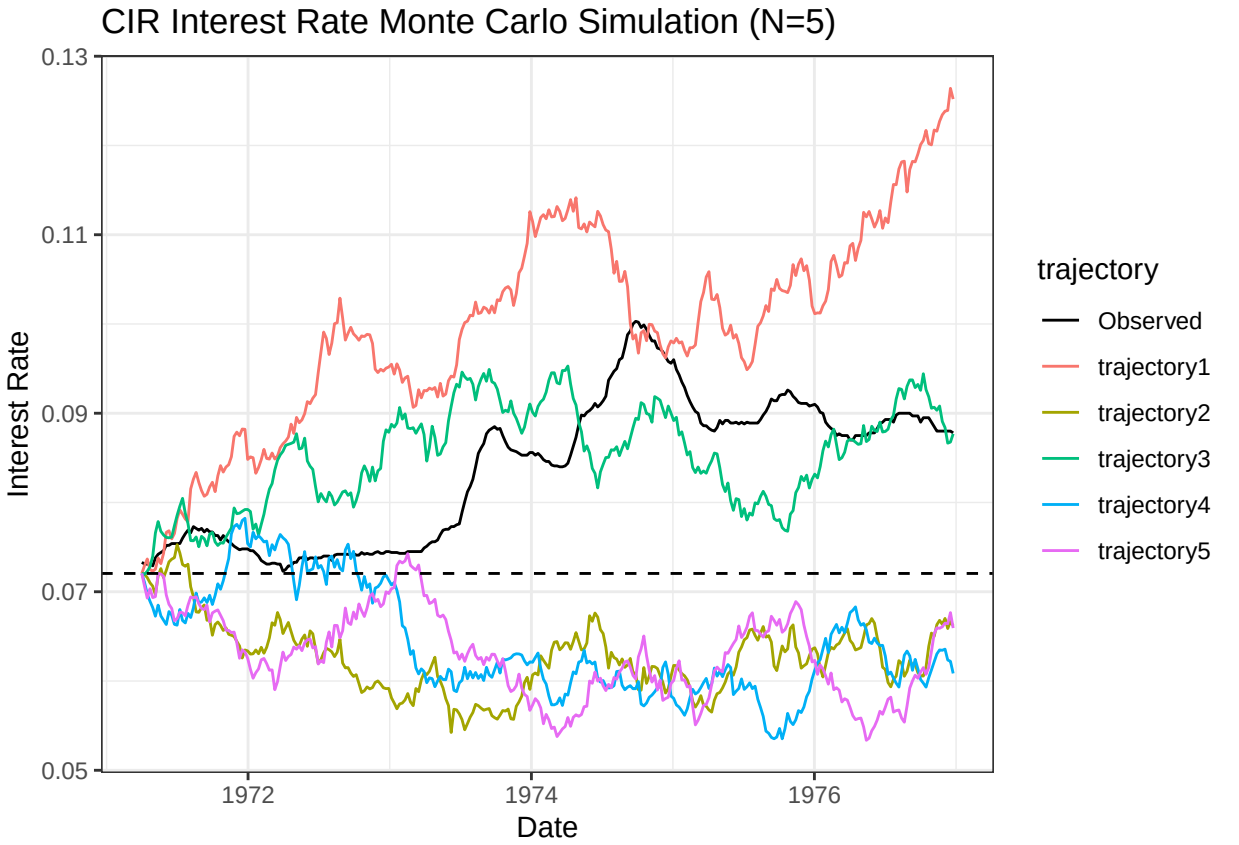
```
r = usrates$rate/100
n = length(r)
rates <- data.frame(date = usrates$date, rate = r)

# predicting last n-1 rates based on the first n-1 to fit CIR Model
x <- r[-n]
y = r[-1]/sqrt(x)

X <- matrix(c(x/sqrt(x), rep(1, n-1)/sqrt(x)), ncol=2)
B <- solve(t(X)%*%X)%*%t(X)%*%y
e <- y - X%*%B # for calculating standard error
# r[t]/sqrt(r[t-1]) ~ a*r[t-1]/sqrt(r[t-1]) + b*(1/sqrt(r[t-1])) + ep
# a = (1 - alpha*dt), b = alpha*theta*dt, ep = sigma*dW
```

```
a <- B[1]
b <- B[2]
MSE <- (t(e)%*%e)[1,1]/(n-3)

dt <- 1 # time step
# initial parameters from least squares regression
alpha = (1 - a)/dt # Long-term mean or equilibrium interest rate
theta = b/(1 - a) # Speed of mean reversion
sigma <- sqrt(MSE)/sqrt(dt) # Volatility
```

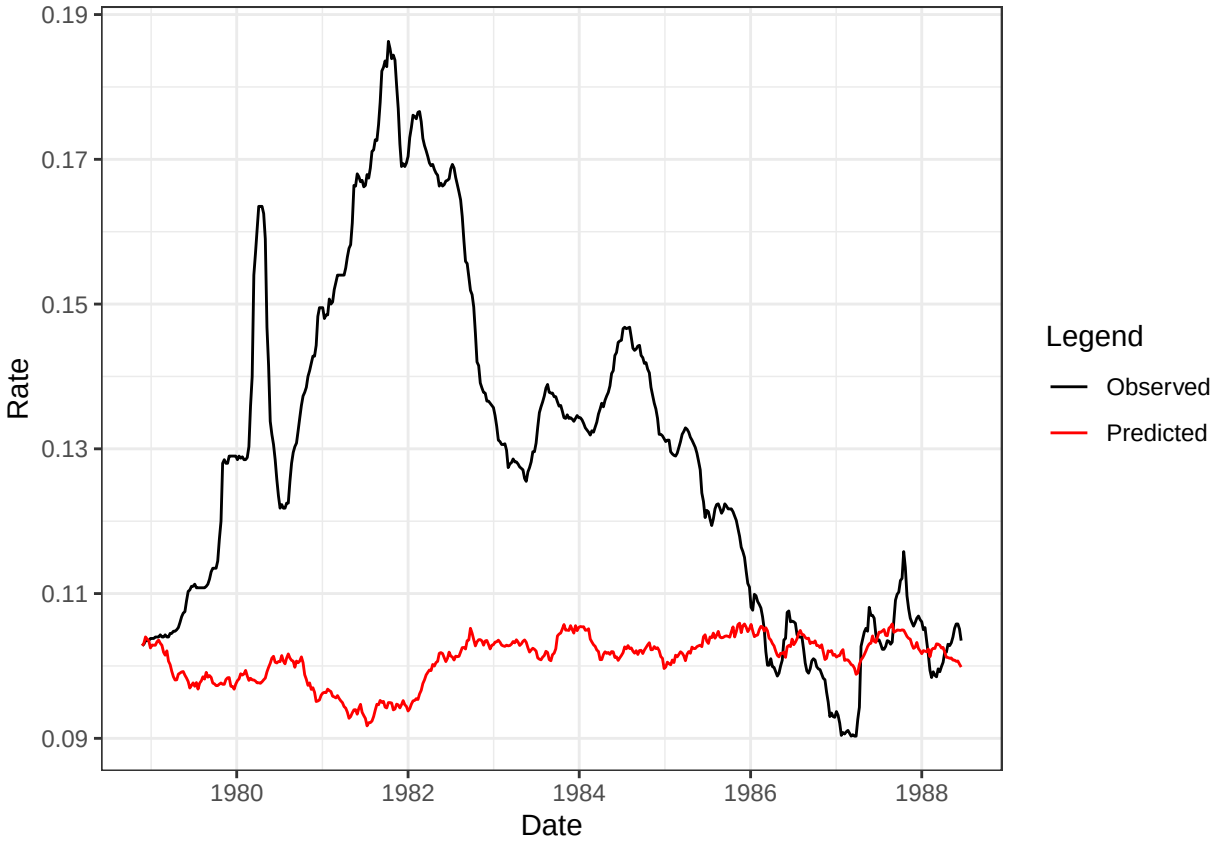
There are two ways to calculate the Log-Likelihood for the CIR model outlined by (Kladivko 2007) both of which are implemented in the file `functions.R` (though I am using the "Bessel" version here).

```
# Now refine the value of parameters with ML estimation
logLikelihood <- function(param) {
  result <- lnLhelper(param=param, data=r, dt=1)
  return(result)
}

# optimize using optim()
MLE <- optim(c(alpha, theta, sigma), logLikelihood)$par
```

Now that we have chosen parameters, we can generate samples.




Now we would like to quantify how well the model characterizes the observed rates. One way to do this is to generate several possible trajectories, and see how far away the average is from the observed rates.



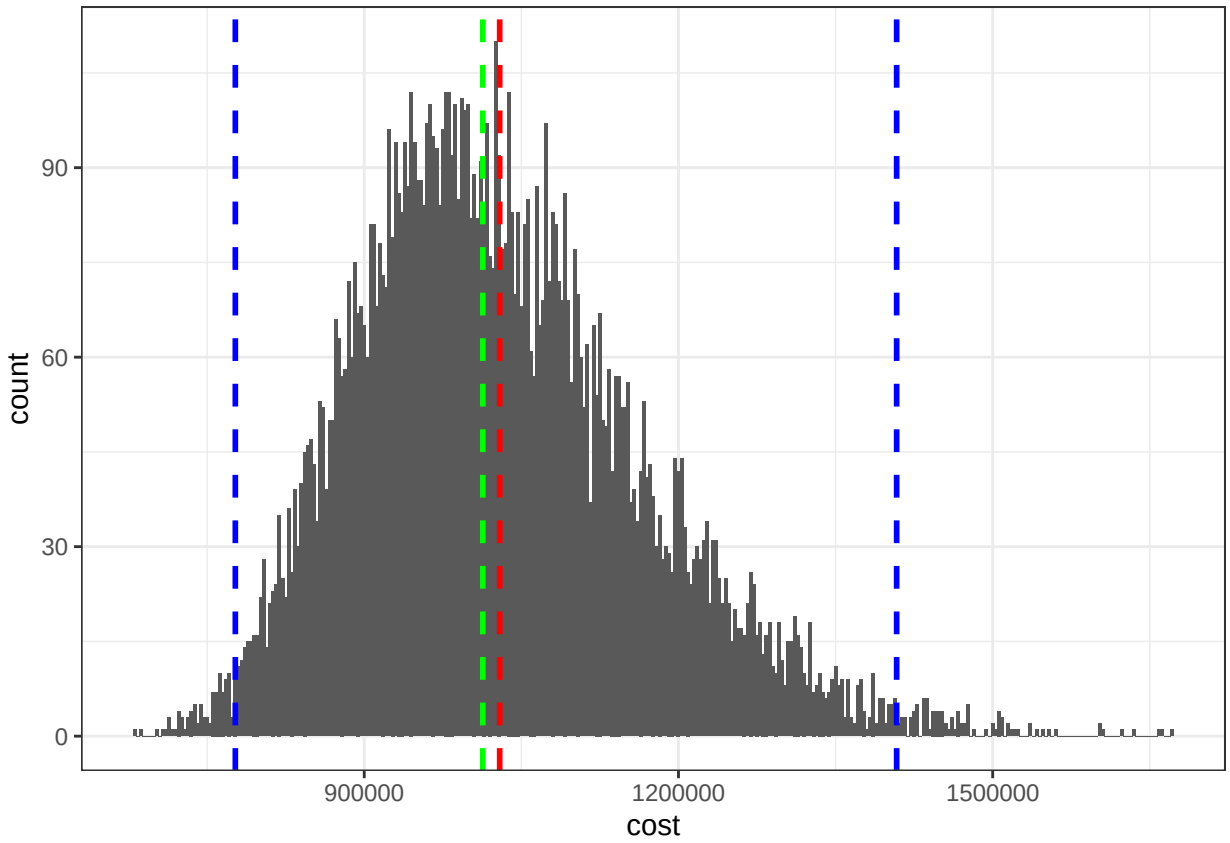

We can increase the number of CIR trajectories we include in our average to get a prediction with less variance but higher bias. It's fun to play around with different intervals and iteration numbers to see the kind of predictions we get.

In the end, our predictions aren't actually very good by this metric. But it really feels like there's something here. If you generate the plot where multiple trajectories are shown alongside the observed values several times, it seems like maybe we should be able to make statements like "the observed interest rate is bounded from above and below by the $Q_u$, $Q_\ell$ upper and lower quantiles 90% of the time", or something like this.

## Comparing mortgaging and renting

Let's assume that the interest rate trajectories generated by the preceding method are reasonable. What can we say about the likely cost of a mortgage? And can we compare that in a meaningful way to the cost of renting?

First of all, let's generate $10,000$ interest rate trajectories. Our data was weekly, but it's easier to compute costs from monthly data, so we need to agregate the data. We can compute the total cost of a 25 year mortgage (on a $50000 property) on each trajectory and look at the resulting histogram.




```
##         1%       50%       99%
##  776938.3 1013199.4 1408284.5
```

The blue lines are the 0.01 and 0.99 quantiles, the green line is the median, and the red line is the mean.

If we consider the "true" value of a home to be the principal value it was sold for; then we see that even in the best case a mortgager will still overpay by over 50%. On average they will overpay by over double.

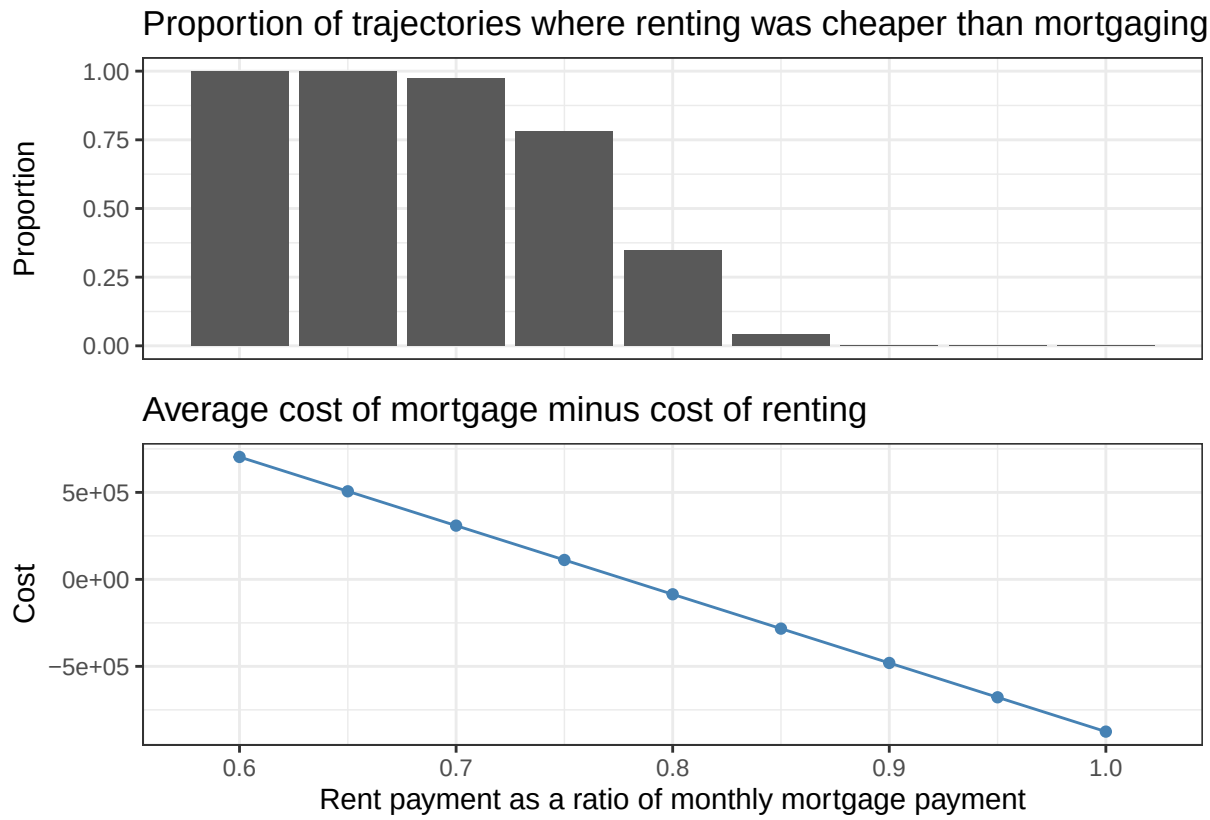
When I first saw this I was floored. Maybe I'm just a bit naive but that is an absurd amount of money that goes towards interest. But we have to live somewhere, so how does this compare to renting? This is actually not a trivial comparison to make. We'll have to make some assumptions.

1. Rental prices are proportional to the value of the property at the start of the lease term
2. Property value appreciates at a rate of $\alpha = 0.02$
3. Stocks/other holdings appreciate at a rate of $\beta = 0.05$

With these assumptions we can calculate the cost of renting a property from the mortgage cost. Let's go through an example. A house is bought in 1990 for $\$500,000$. What should the rent be set at? We use the interest rate in 1990 (say 0.03) and the property value to compute the monthly payment on a fixed rate 25 year mortgage. The rent should be set to some percentage of this monthly payment (maybe 0.6?). Then in 1991, the property value appreciates by 2% so the new value is \$51000. We look at the interest rate in the 1991 and compute the monthly payment on a fixed rate mortgage. Then the rent for the year of 1991 will be some proportion of that monthly payment. It's not clear what proportion of the mortgage cost should rent be set to, so we can check a range of possibilities.

There is another assumption. Any money that our hypothetical renter saved on the monthly cost of rent compared to the monthly cost of the mortgage is being invested in the market in stocks/other holdings. This is most likely not true for the average renter.

Below are two plots and a table. The first plot shows the proportion of rates where renting came out on top for each monthly rent as a ratio of monthly mortgage payment. Unsurprisingly, as the cost of rent approaches the cost of a mortgage, it becomes better to mortgage in more cases. The second plot shows the average difference between the cost of renting and the cost of mortgaging. The table is a concise summary of both plots.




| ratio | proportion | mean |
|---|---|---|
| 0.60 | 0.9999 | 703562.06 |
| 0.65 | 0.9990 | 506213.71 |
| 0.70 | 0.9740 | 308865.35 |
| 0.75 | 0.7818 | 111517.00 |
| 0.80 | 0.3486 | -85831.36 |
| 0.85 | 0.0432 | -283179.72 |
| 0.90 | 0.0002 | -480528.07 |
| 0.95 | 0.0000 | -677876.43 |
| 1.00 | 0.0000 | -875224.78 |

What this graph is really showing is that the trade-off between renting and buying comes down to whether or not we think housing will appreciate in value more than other holdings. NOT because owning the house is good for the home owner, but because not owning is bad for the renter. Even if you own your home and the price sky rockets, if you wanted to sell to take advantage you would then just end up buying a house in the inflated market. In my opinion what buying really does is lock in a price at the current market rate. Someone

who rents is constantly paying proportional to the current price of housing, while the mortgager pays monthly mortgage payments proportional to the market rates at the time of when they bought. However, if other holdings grow faster than housing the renter can still come out on top by investing what they would have put towards their mortgage into those other holdings.

## Conclusions

There are several issues with this project. First of all I feel that the CIR model is better suited to predicting over shorter intervals than the 25 years that I asked of it. Secondly, I am forced to make assumptions about the way that rentals are priced, and those assumptions shaped my results. Had I chosen different parameter $\alpha$ and $\beta$ (appreciation of property and other holdings), I would have gotten different results.

However, I still think this project was worth doing. It allowed me to practice manipulating data, and it was fun learning about and implementing stochastic volatility models. I would also say that even though the conclusions should be taken with a grain of salt, I do think they hold some value.

## Bibliography

Kladivko, Kamil. 2007. "Maximum Likelihood Estimation of the Cox–Ingersoll–Ross Process: The Matlab Implementation," January.

Mac, Freddie. 2016. "30-Year Fixed Rate Mortgage Average in the United States [MORTGAGE30US], Retrieved from FRED, Federal Reserve Bank of St. Louis; Https://Fred.stlouisfed.org/Series/MORTGAGE30US, October 16, 2024."