Lab 3

2/8/2022

Housing Data

In class, we fit a few models using the housing data that we've been considering in lecture. In lab, we'll take a deeper dive into the data set. First, let's load the data

```
fileName <- url("https://raw.githubusercontent.com/ysamwang/btry6020_sp22/main/lectureData/estate.csv")
housing_data <- read.csv(fileName)

head(housing_data)
```

```
##   id  price area bed bath  ac garage pool year quality style   lot highway
## 1  1 360000 3032   4    4 yes      2   no 1972  medium     1 22221      no
## 2  2 340000 2058   4    2 yes      2   no 1976  medium     1 22912      no
## 3  3 250000 1780   4    3 yes      2   no 1980  medium     1 21345      no
## 4  4 205500 1638   4    2 yes      2   no 1963  medium     1 17342      no
## 5  5 275500 2196   4    3 yes      2   no 1968  medium     7 21786      no
## 6  6 248000 1966   4    3 yes      5  yes 1972  medium     1 18902      no
```

Recall that there are 522 observations with the following variables:

- price: in 2002 dollars
- area: Square footage
- bed: number of bedrooms
- bath: number of bathrooms
- ac: central AC (yes/no)
- garage: number of garage spaces
- pool: yes/no
- year: year of construction
- quality: high/medium/low
- home style: coded 1 through 7
- lot size: sq ft
- highway: near a highway (yes/no)

There is no age data in the table, but we can compute it on our own from the year variable

```
housing_data$age <- 2002 - housing_data$year
```

Polynomial regression

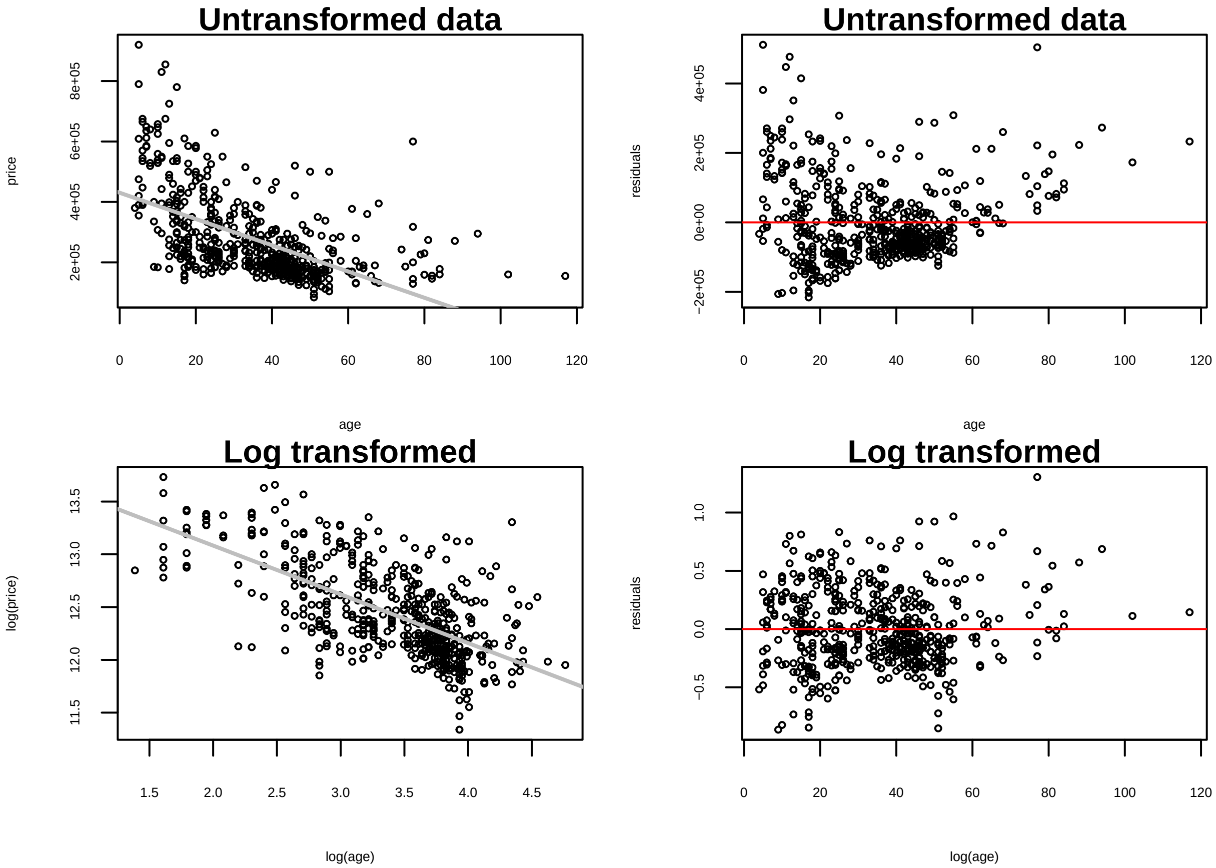
Taking the log of price and the log of age in order make the data look more linear. To compare, we fit a linear model to both the raw data and the log transformed data

```
reg_linear <- lm(price ~ age, data = housing_data)
reg_log <- lm(log(price) ~ log(age), data = housing_data)



par(mfrow = c(2,2), mar = c(4, 4, 1, 1))
plot(housing_data$age, housing_data$price, cex.lab = .5, cex.axis = .5,
     cex = .5, main = "Untransformed data", xlab = "age", ylab = "price")
abline(a = reg_linear$coef[1], b = reg_linear$coef[2], col = "gray", lwd = 2)
plot(housing_data$age, reg_linear$res, cex.lab = .5, cex.axis = .5,
     cex = .5, main = "Untransformed data", xlab = "age", ylab = "residuals")
abline(h = 0, col = "red")

plot(log(housing_data$age), log(housing_data$price), cex.lab = .5, cex.axis = .5,
     cex = .5, main = "Log transformed", xlab = "log(age)", ylab = "log(price)")
abline(a = reg_log$coef[1], b = reg_log$coef[2], col = "gray", lwd = 2)
plot(housing_data$age, reg_log$res, cex.lab = .5, cex.axis = .5,
     cex = .5, main = "Log transformed", xlab = "log(age)", ylab = "residuals")
abline(h = 0, col = "red")
```



We can see that the log transformed data looks much better, but as an alternative to fitting the transformed regression, we could also use polynomial regression instead of transforming the data. Let's use the raw data, but also include a covariate of age squared.

```
## R requires you to use I(age^2) instead of just including age^2
reg_quad1 <- lm(price ~ age + I(age^2), data = housing_data)
summary(reg_quad1)
```

```
##
## Call:
## lm(formula = price ~ age + I(age^2), data = housing_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -280265  -55366  -21785   49671  432273
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 552349.97   15860.35   34.83   <2e-16 ***
## age         -11922.03     795.81  -14.98   <2e-16 ***
## I(age^2)        93.34       9.26   10.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 105100 on 519 degrees of freedom
## Multiple R-squared:  0.4218, Adjusted R-squared:  0.4196
## F-statistic: 189.3 on 2 and 519 DF,  p-value: < 2.2e-16
```

The variables, age and age squared will be quite correlated, which as we will see on Wednesday can be a bad thing. So we typically will want to use a transformation of the polynomial covariates which are not as highly correlated. We will use the `poly` function which takes the covariate and the degree of the polynomial (in this case 2) and return a set of covariates which act like age and age squared, but are not correlated. It's also easier to type out instead of including a bunch of terms by hand. The coefficients aren't directly interpretable since the covariates aren't exactly age and age squared anymore, but we can see that they give the same fitted values as before.

```r
reg_quad2 <- lm(price ~ poly(age,2), data = housing_data)
summary(reg_quad2)
```

```
##
## Call:
## lm(formula = price ~ poly(age, 2), data = housing_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -280265  -55366  -21785   49671  432273
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      277894       4599   60.42   <2e-16 ***
## poly(age, 2)1 -1748855     105079  -16.64   <2e-16 ***
## poly(age, 2)2  1059186     105079   10.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 105100 on 519 degrees of freedom
## Multiple R-squared:  0.4218, Adjusted R-squared:  0.4196
## F-statistic: 189.3 on 2 and 519 DF,  p-value: < 2.2e-16
```

```r
sum(abs(reg_quad1$fitted.values - reg_quad2$fitted.values))
```

```
## [1] 1.193257e-08
```

We can't directly compare the R^2 of log transformed model to the quadratic model since the dependent variables are different, but we can take the `exp()` of the fitted values of the log model to bring the units back to dollars. We can then compare the RSS of the log model, the linear model, and the model which includes the quadratic term:

```r
sum((housing_data$price - exp(reg_log$fitted.values))^2)
```

```
## [1] 5.760537e+12
```

```
sum((housing_data$price - reg_linear$fitted.values)^2)
```

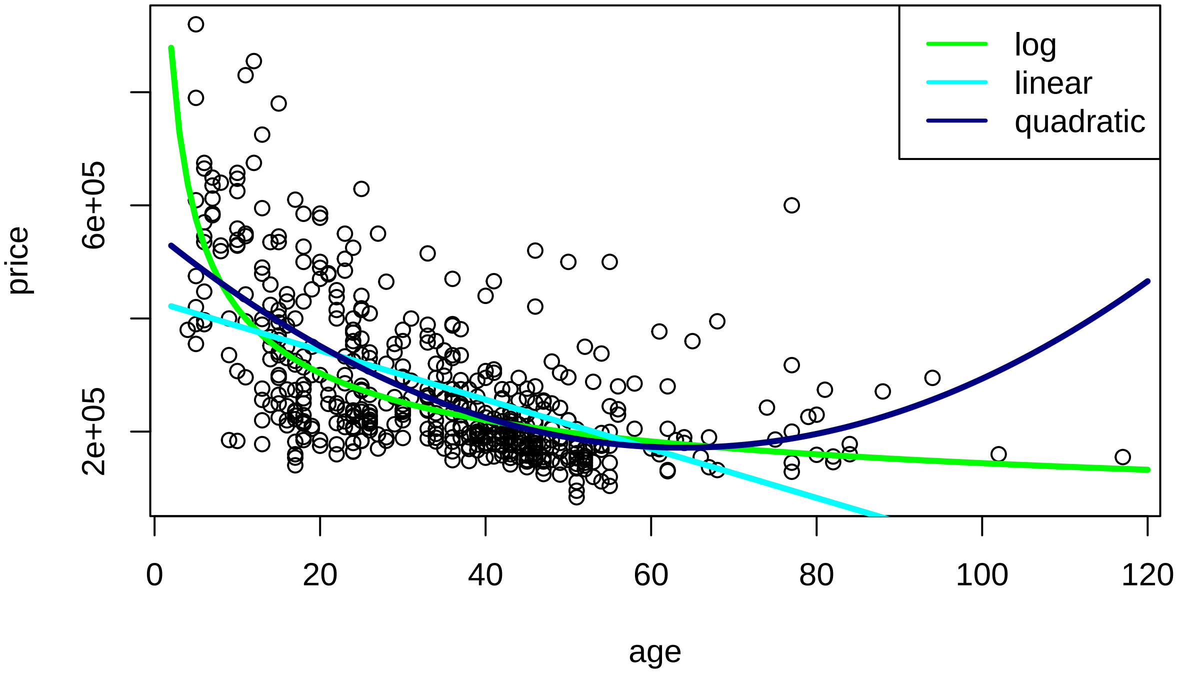
```
## [1] 6.852419e+12
```

```
sum((housing_data$price - reg_quad2$fitted.values)^2)
```

```
## [1] 5.730544e+12
```

We can also plot the fitted prices for each model. For this, we will use the `predict` function. The `predict` function takes an `lm` object and a data frame of covariate observations. It then computes the predicted value of the covariate observations based on the coefficients estimated in the `lm` object. Note that for the log regression, we still feed `predict` the raw (i.e., untransformed ages) but the predicted values are in log(price).

```
plot(housing_data$age, housing_data$price, xlab = "age", ylab = "price")
lines(2:120, exp(predict(reg_log, data.frame(age = 2:120))),
      col = "green", lwd = 3)
lines(2:120, predict(reg_linear, data.frame(age = 2:120)),
      col = "cyan", lwd = 3)
lines(2:120, predict(reg_quad1, data.frame(age = 2:120)),
      col = "navy", lwd = 3)
legend("topright", col = c("green", "cyan", "navy"),
       legend = c("log", "linear", "quadratic"), lwd = 2)
```



Question:

- We can see that the model for the log transform drastiacally outperforms the model with only the linear model. However, the model with the quadratic term slightly outperforms the model with the log transformed data. With your neighbors, discuss which model you would use if you were fitting the data?
- What if you were trying to explain this model to a collaborator?
- What if you were just trying to predict what you should sell your house for?
- What if the house you are selling is 150 years old?

Can we improve the quadratic model? Let's see if we can just fit higher polynomials to the data. Using a 3rd degree polynomial is called a cubic and using a 4th degree polynomial is called a quartic.

```
reg_cubic <- lm(price ~ poly(age,3), data = housing_data)
reg_quartic <- lm(price ~ poly(age,4), data = housing_data)

summary(reg_quad2)
```

```
##
## Call:
## lm(formula = price ~ poly(age, 2), data = housing_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -280265  -55366  -21785   49671  432273
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     277894       4599   60.42   <2e-16 ***
## poly(age, 2)1 -1748855     105079  -16.64   <2e-16 ***
## poly(age, 2)2  1059186     105079   10.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 105100 on 519 degrees of freedom
## Multiple R-squared:  0.4218, Adjusted R-squared:  0.4196
## F-statistic: 189.3 on 2 and 519 DF,  p-value: < 2.2e-16
```

```
summary(reg_cubic)
```

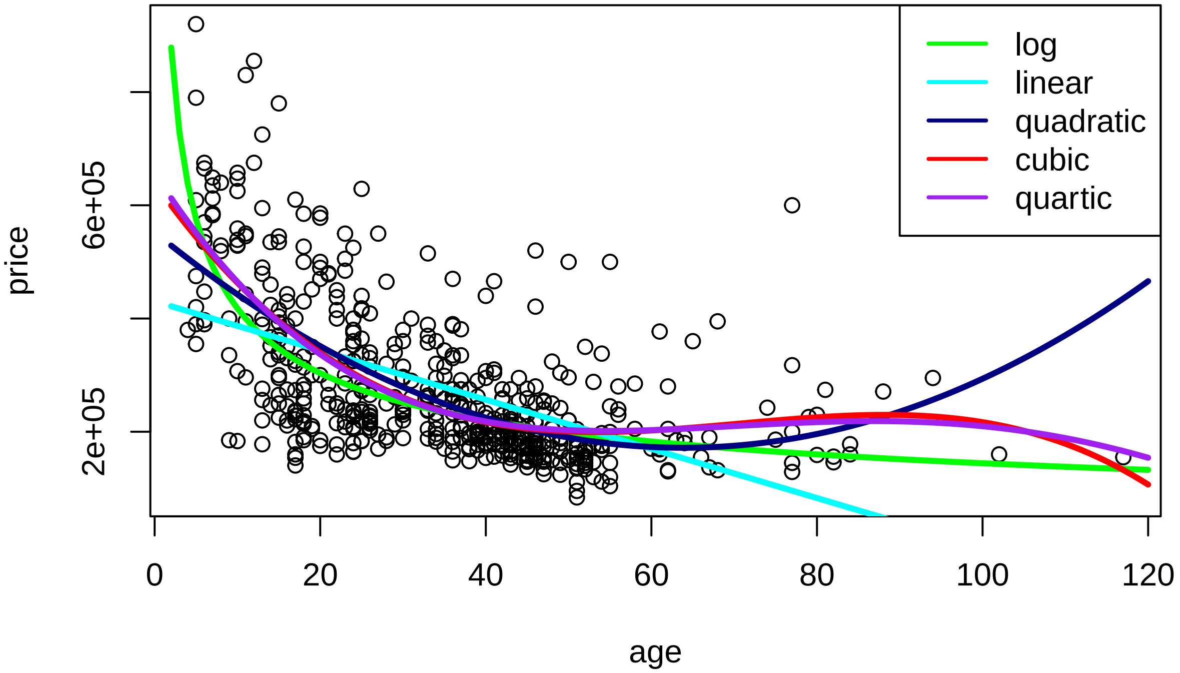
```
##
## Call:
## lm(formula = price ~ poly(age, 3), data = housing_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -293191  -55948  -22012   47322  420616
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     277894       4502  61.728  < 2e-16 ***
## poly(age, 3)1 -1748855     102857 -17.003  < 2e-16 ***
## poly(age, 3)2  1059186     102857  10.298  < 2e-16 ***
## poly(age, 3)3  -500361     102857  -4.865 1.52e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 102900 on 518 degrees of freedom
## Multiple R-squared:  0.4471, Adjusted R-squared:  0.4439
## F-statistic: 139.6 on 3 and 518 DF,  p-value: < 2.2e-16
```

```
summary(reg_quartic)
```

```
##
## Call:
## lm(formula = price ~ poly(age, 4), data = housing_data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -295682  -56477  -22001   47865  420627
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      277894       4504  61.696  < 2e-16 ***
## poly(age, 4)1 -1748855     102910 -16.994  < 2e-16 ***
## poly(age, 4)2  1059186     102910  10.292  < 2e-16 ***
## poly(age, 4)3  -500361     102910  -4.862 1.54e-06 ***
## poly(age, 4)4    70294     102910   0.683    0.495
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 102900 on 517 degrees of freedom
## Multiple R-squared:  0.4476, Adjusted R-squared:  0.4433
## F-statistic: 104.7 on 4 and 517 DF,  p-value: < 2.2e-16
```

```
plot(housing_data$age, housing_data$price, xlab = "age", ylab = "price")
lines(2:120, exp(predict(reg_log, data.frame(age = 2:120))),
      col = "green", lwd = 3)
lines(2:120, predict(reg_linear, data.frame(age = 2:120)),
      col = "cyan", lwd = 3)
lines(2:120, predict(reg_quad1, data.frame(age = 2:120)),
      col = "navy", lwd = 3)
lines(2:120, predict(reg_cubic, data.frame(age = 2:120)),
      col = "red", lwd = 3)
lines(2:120, predict(reg_quartic, data.frame(age = 2:120)),
      col = "purple", lwd = 3)
legend("topright", col = c("green", "cyan", "navy", "red", "purple"),
       legend = c("log", "linear", "quadratic", "cubic", "quartic"), lwd = 2)
```



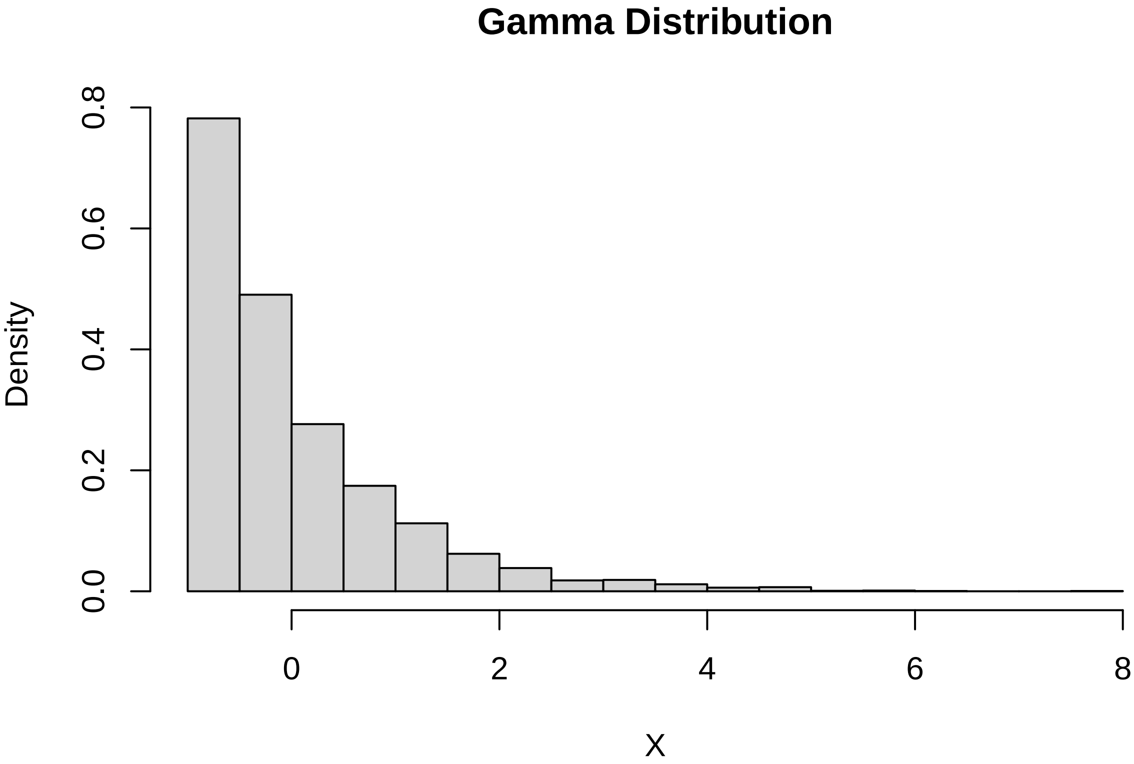
Question:

- Examine the R^2 values for each of the models. Each time we fit a higher order polynomial, the R^2 increases. Will this always be the case or is it just a coincidence? Why do you think so?
- How would you decide which model to use?

Sampling Distributions

We will use a small simulation study to examine how the sampling distributrion of estimated coefficients changes. In particular, we will simulate many different data sets and record the estimated $\hat{b}_1$ each time. We can then look at the distribution of the estimated $\hat{b}_1$, and by changing certain features of the data generating process, we can see how it effects the distribution of the resulting $\hat{b}_1$.

Below, we draw `Y.norm` as a linear model where all the coefficients are 1 except the intercept which is set to 0. The errors are drawn from a normal distribution with variance 1. We draw `Y.gamma` as a linear model where all the coefficients are 1 except the intercept which is set to 0. The errors are drawn from a gamma distribution with variance 1 which is centered to have mean 0. As you can see from the plot below, the gamma distirbution is skewed and not close to a normal distribution.



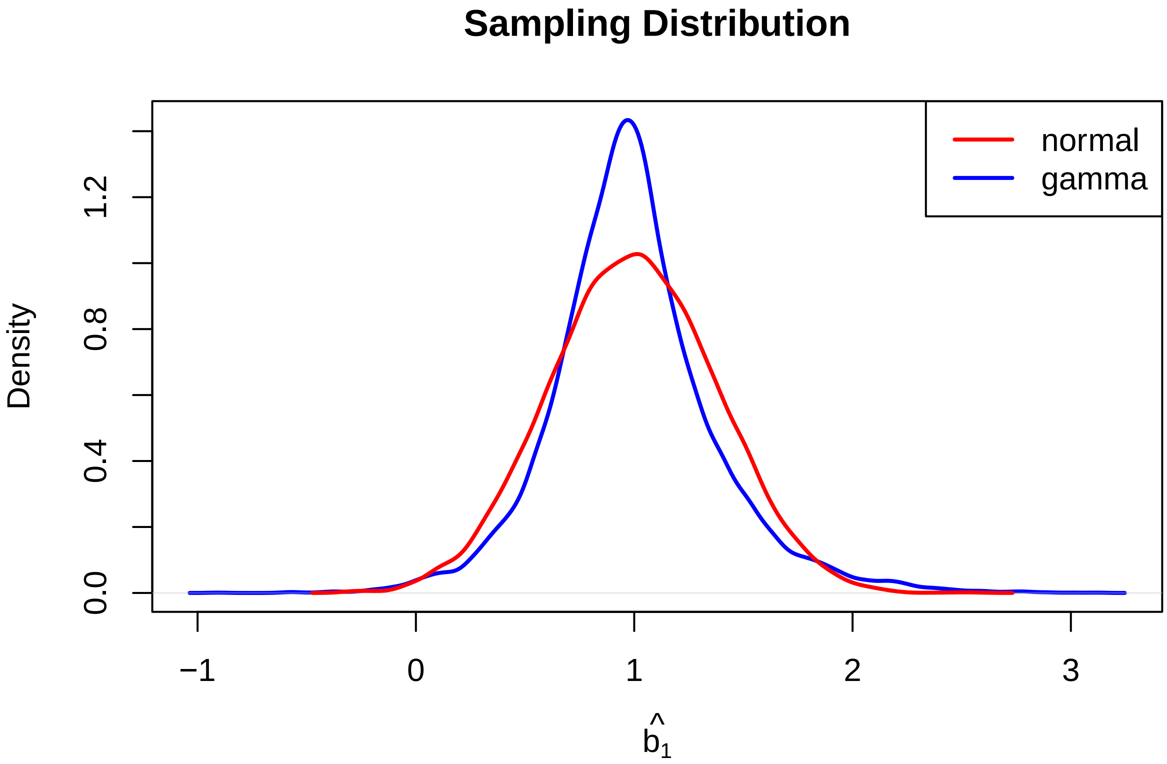
```r
# Number of times we will simulate a new data set
sim.size <- 10000

# number of observations
n <- 8
# number of covariates
p <- 1
# standard deviation of the X values
x.sd <- 1
# drawing the covariates from a normal distribution
X <- matrix(rnorm(n * p, sd = x.sd), n, p)
# coefficients are all set to 1 (except no intercept)
beta <- rep(1, p)

# recording the estimated b_1 for each simulated data set
rec <- matrix(0, sim.size, 2)

for(i in 1:sim.size){
  Y.norm <- X %*% beta + rnorm(n)
  Y.gamma <- X %*% beta + (rgamma(n, 1, 1) - 1)
  reg_norm <- lm(Y.norm ~X)
  reg_gamma <- lm(Y.gamma ~X)
  rec[i, ] <- c(reg_norm$coef[2], reg_gamma$coef[2])
}

# Plotting the histogram of the estimated b_1
plot(density(rec[, 2]), xlab = expression(hat(b[1])),
     type = "l", col = "blue", main = "Sampling Distribution", lwd = 2)
lines(density(rec[, 1]), col = "red", lwd = 2)
legend("topright", col = c("red", "blue"), legend = c("normal", "gamma"), lwd = 2)
```



Questions

- What is the mean of the distribution of $\hat{b}_1$ in each case?
- How do the variances of the two distributions (normal errors vs gamma errors) compare? How do the shapes compare?
- Increase x.sd from 1 to 3 and re-run the simulation. This will cause the X values to be more spread out. What happens to to the distributions of $\hat{b}_1$?
- Change x.sd back to 1 and set n to 50. What happens to to the distributions of $\hat{b}_1$? How do the variances of the two distributions (normal errors vs gamma errors) compare? How do the shapes compare?

Potential Answers to the discussed questions

- We can see that the model for the log transform drastically outperforms the model with only the linear model. However, the model with the quadratic term slightly outperforms the model with the log transformed data. With your neighbors, discuss which model you would use if you were fitting the data? What if you were trying to explain this model to a collaborator? What if you were just trying to predict what you should sell your house for? What if the house you are selling is 150 years old?
  - It is true that the quadratic model performs better in terms of RSS than the log model, however, it might be easier to explain to a collaborator For instance, it could be easier to just say a 1% difference in square footage results in some corresponding % difference in price rather than giving the interpretation of the polynomial model for which a corresponding statement about the difference in price corresponding to a difference in square footage depends on the specific value of square footage being considered. If you really are just concerned about prediction though, the smaller RSS of the quadratic model suggests that it might do better in prediction. However, the quadatic model is a more complicated than the log model (in the sense that it includes 2 covariates instead of 1) so it might be the case that we've overfit to the observed data and it's not clear whether the quadratic model would be better at predicting in new data. We also see that the quadratic model becomes a pretty bad fit for larger values of age, so we would need to be careful when predicting the price of older homes.
- Examine the R^2 values for each of the models. Each time we fit a higher order polynomial, the R^2 increases. Will this always be the case or is it just a coincidence? Why do you think so? How would you decide which model to use?
  - Yes, the R^2 will always increase if you increase the degree of the polynomial. This is because when we fit the higher order polynomial, the best line fitting line of a lower order polynomial is always an option because we can set the coefficients of the higher degrees to be 0. So any best fitting line for the higher order polynomial must be at least as good as the best fitting line for the lower fitting polynomial. The cubic and quartic models do seem to have better behavior when predicting the price of older homes. However, they are even harder to interpret and explain. Also, it might be the case that we've overfit to the observed data and they may not be better at predicting in new data. Also, the difference between the R^2 of the cubic and quartic models is very small so the additional complexity of the quartic model might not be worth it.