STAT 491 - Lecture 11: Bayesian Regression

Generalized Linear Models

Generalized linear models (GLMs) are a class of techniques that include linear regression, logistic regression, and Poisson regression. There are three defining features for a GLM:

1. A sampling model,

2. A linear combination of predictors, and

3. A link function.

Variable types and sampling models

In a GLM setting there are two sets of variables:

1. *Predicted Variable:* also known as the dependent variable and usually denoted by y. The goal of the analysis is to create a model for the predicted variable.

2. *Predictor Variable:* also known as the independent variables and are usually denoted by X. This set of variables are used to predict the dependent variable.

- When constructing a GLM, we need to think about the sampling model, $p(y|X, \theta)$. The sampling model is now conditional on not only the unknown parameters, θ, but also the set of predictor variables, X.

- Scale type of variables. The structure of the sampling model will be largely dependent on the measurement scale of the predicted variable. Consider a set of athletes participating in a mountain bike race. The results could be reported as a time for completing the race, place in the race, or team participating on. These three different results are examples of metric, ordinal, and nominal scales.

  - metric: time for the race is a metric response. Metric data are measured on an interval or ratio scale. A special subset of metric data consists of *count*, or frequency, data. Count data are different from other metric data in that the data are non-negative and not continuous.

  - ordinal: the place in the race is an example of an ordinal response. Ordinal data has an ordered structure, but lacks additional information about the differences in each level. For instance, was the difference between first and second place one second or one hour? It is impossible to tell. Another example would be Likert style data.

  - nominal: like ordinal data, nominal data is a discrete categorical variable. However, nominal data does not have an inherent ordering. Examples would be team of the mountain biker or hair color.

- The variable type of the predicted variable will ultimately have a sampling distribution, but we will talk about this later.

Linear Combinations of Predictors

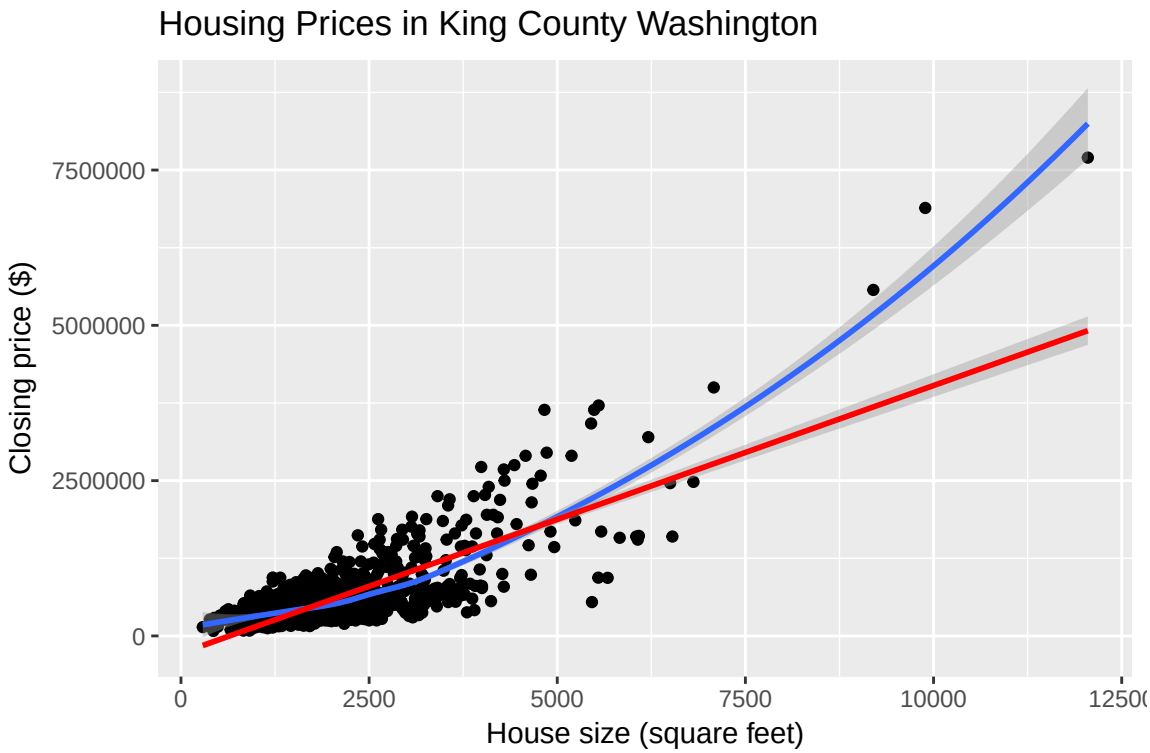
The main element of linear modeling is a relationship between predictors and the predicted variable.

```
SeattleHousing <- read.csv('http://math.montana.edu/ahoegh/teaching/stat408/datasets/SeattleHousing.csv
```

Let the predicted, or dependent variable, be the price of the home sale denoted as y.

Metric Predictors

- *Linear function of a single metric predictor*: Initially consider a single metric variable as the predictor, say square footage of the home.
  - A linear relationship is the element of a linear modeling and carries the assumption that there is a proportional relationship between the dependent and independent variable.
  - The general mathematical form for a linear function is $y = \beta_0 + \beta_1 x$, where β_0 is an intercept term and β_1 is the slope parameter. The coefficent β_1 controls how much y increases based on a one unit increase in x.
  - If we plot values y and x we should see a straight line.



- In this case, the figure suggests that a linear relationship might not be the best representation of the data. So what do we do?? One option is to add polynomial terms of square footage.

- *Additive combination of metric predictors:* often multiple predictors can be used for prediction. Here consider also using bedrooms and bathrooms as predictor variables. Note we can also use a squared term for House size in this same framework.

  - Consider predicting, y, with with a generic set of predictors, x, according to $y = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$.

  - An assumption of this model is that the predictors have an additive relationship, meaning that an increase in one predictor variable will result in the same increase in the predicted variable for any value of the other parameters.

  - Non linear effects can be modeled using this framework with a polynomial basis function. For instance consider $price = \beta_0 + \beta_1 x_{sqft} + \beta_2 x_{sqft}^2$.

- *Nonadditive combination of metric predictors:* The combined influence of two parameters does not have to be additive. There can be an interaction effect between two parameters. For instance, consider zipcode and square footage relative to price. The additive relationship assumes that each additional foot of space in the house has the same increase in house price for each zipcode.

  - A multiplicative combination of predictors captures *interactions* between terms and can be formulated as $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 \times x_2$.

  - Higher level interactions will add many parameters to the model and can make interpreting the results difficult.

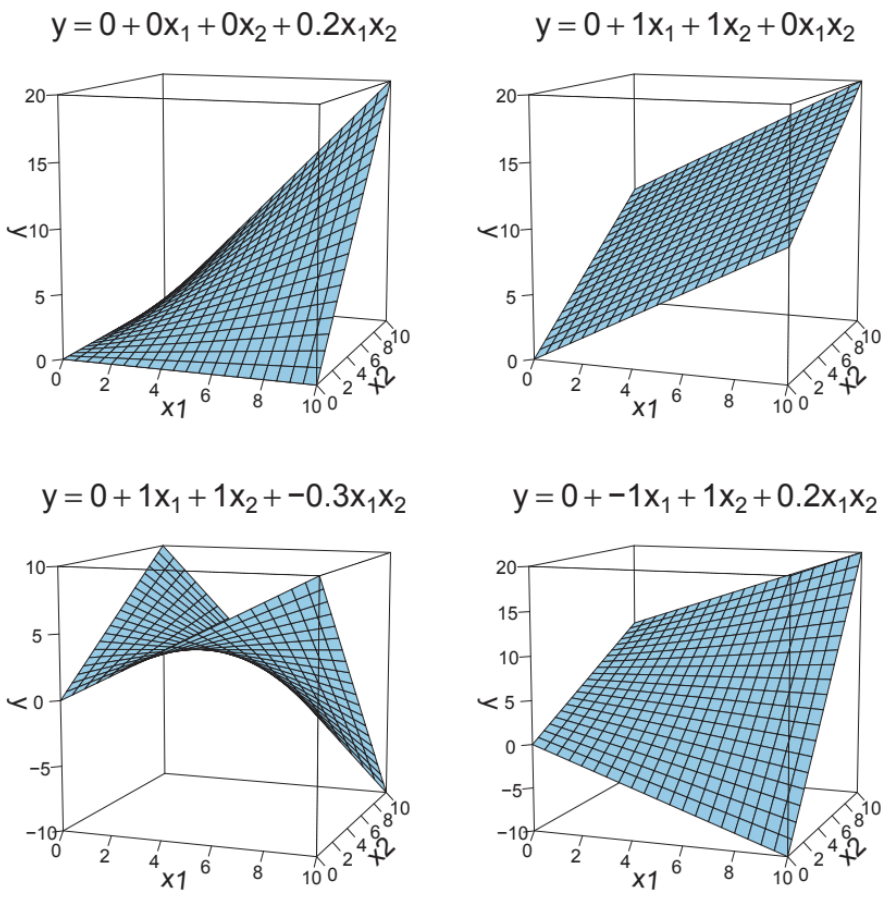


Figure 15.3: Multiplicative interaction of two variables, x_1 and x_2. Upper right panel shows *zero* interaction, for comparison. Figure 18.8, p. 502, provides additional perspective and insight. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

Figure 1: Multiplicative Interaction

Nominal Predictors

Similarly we can consider nominal predictors in a linear model.

- *Linear model for a single nominal predictor:* In this case the nominal predictor has a step or deflection on the predicted variable. For instance housing prices for Bozeman, Livingston, and Belgrade. If the home is in Bozeman, there will be a higher expected price.

- Let the levels of the nominal variable be denoted as $\tilde{x} = \{x_{[1]}, \ldots, x_{[L]}\}$. With this notation, $x_{[1]} = 1$ if the observation comes from group 1 and 0 otherwise. These variables are known as dummy variable.

- Then $y = \beta_0 + \beta_{[1]}x_{[1]} + \ldots \beta_{[L]}x_{[l]}$.

- Using nominal values in a linear model requires a constraint on the values. Two common options are to set $\beta_0 =$, then the $\beta_{[i]}$ values represent group means, say for a zipcode. Another option would be to constrain $\sum_j = 1^L \beta_{[L]} = 0$.

- *Additive and Nonadditive combinations of mixed-type predictors:* The same principle applies as before when looking at combinations of nominal predictors as well as combinations of nominal and metric predictors.

Linking combined predictors to predicted data

- Given a linear combination of predictors, the final step is to map that relationship to the predicted variable.

- Formally, this mapping using a link function. Let $f()$ be an inverse link function, then $\mu = f(X\beta)$, where $X\beta = \sum x_i \beta_i$ is the the linear combination of predictors and μ is the central tendency of the predicted variable.

- An inverse link function maps the predictors to the same 'space' or support as the predicted variable.

- In a traditional linear model (multiple linear regression), the link function is simply the identify function. That is $f(X\beta) = X\beta$.

- When the predicted variable is a binary outcome, the logistic function is frequently used as the inverse link function and gives name to logistic regression.

$$y = logistic(x\beta) = \frac{1}{1 + \exp(-X\beta)}$$

- The logit function, $logit(p) = log(\frac{p}{1-p})$ is the link function associated witht he inverse-link logistic function.

- The logistic function will map $X\beta \in [-\infty, \infty]$ to $\mu \in [0, 1]$.

- The link function helps to establish the central tendency of the data, but we still need to account for noisy data. In other words, we will use a probability distribution function to map μ to the predicted data y as $y \sim pdf(\mu, ..[\text{other parameters}])$.

Formal Expression of the GLM

The GLM can be written as as two equations:

1. $\mu = f(X\beta)$ and
2. $y \sim pdf(\mu, [\text{parameters}])$

- A table below shows some of the models types that we will explore in this class.

y	$y \sim pdf()$	$\mu = f(X\beta)$
Metric	$y \sim N(\mu, \sigma)$	$\mu = X\beta$
Dichotomous	$y \sim Bernoulli(\mu)$	$\mu = logistic(X\beta)$
Count	$y \sim Poisson(\mu)$	$\mu = \exp(XB)$

Metric Predicted Variable with One Metric Predictor

- Consider predicting a metric variabel with a single metric predictor.

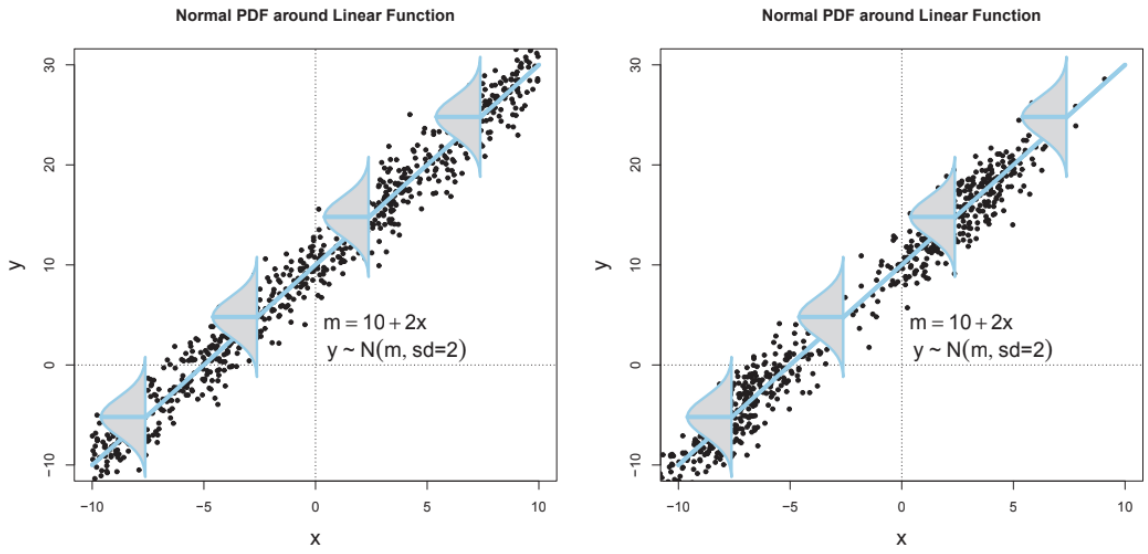


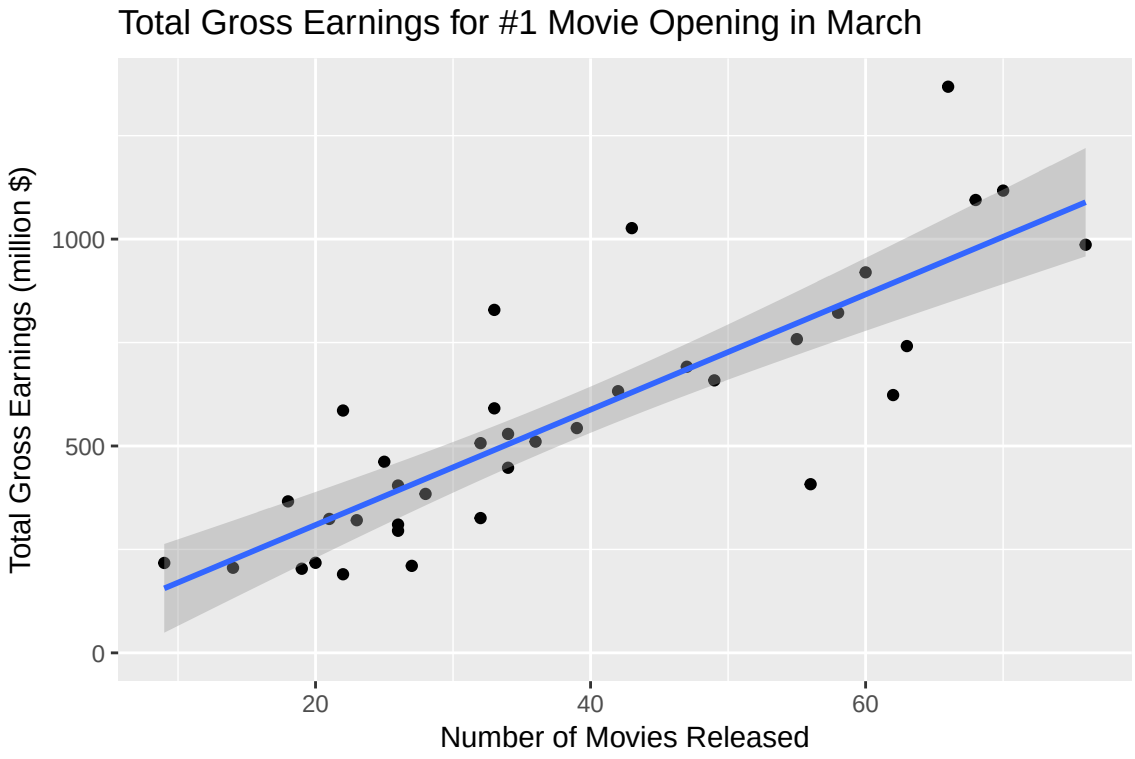
Figure 17.1: Examples of points normally distributed around a linear function. (The left panel repeats Figure 15.9, p. 406.) The model assumes that the data y are normally distributed vertically around the line, as shown. Moreover, the variance of y is the same at all values of x. The model puts no constraints on the distribution of x. The right panel shows a case in which x are distributed bimodally, whereas in the left panel the x are distributed uniformly. In both panels, there is homogeneity of variance.

- Below the code scrapes movie information for the highest earning movie released in March. This dataset contains total earnings and the total number of movies released that month.

- With this model, we are using the identity function as the link to fit a simple linear regression model.

```
#Note this table may update every month
movies <- "http://www.boxofficemojo.com/monthly/"
movie.table <- read_html(movies) %>% html_nodes('table') %>% `[[`(4) %>% html_table() %>% tbl_df()
colnames(movie.table) <- movie.table[1,]
movie.table <- movie.table %>% slice(-1) %>% select(Year, `Total Gross`, Movies, `#1 Movie`)
movie.table$`Total Gross` <- as.numeric(str_replace_all(movie.table$`Total Gross`, "[$,]",''))
movie.table$Movies <- as.numeric(movie.table$Movies)
movie.table
```

```
## # A tibble: 37 × 4
##     Year `Total Gross` Movies                `#1 Movie`
##    <chr>         <dbl>  <dbl>                     <chr>
## 1   2018         407.6     56         A Wrinkle in Time
## 2   2017        1368.3     66      Beauty and the Beast
## 3   2016        1094.5     68                 Zootopia
## 4   2015         758.2     55                Cinderella
## 5   2014         741.5     63                 Divergent
## 6   2013         986.4     76 Oz The Great and Powerful
## 7   2012        1117.2     70          The Hunger Games
## 8   2011         658.6     49                     Rango
## 9   2010        1026.5     43       Alice in Wonderland
## 10  2009         691.5     47       Monsters Vs. Aliens
## # ... with 27 more rows
```



Model Formulation for Bayesian Regression

Nothing that we have talked about is inherently Bayesian, but we can formulate this in a similar framework.

1. **Sampling Model:** In the regression setting $y \sim N(X\beta, \sigma^2)$.

2. **Priors:** What do we need priors on in this model:
   - $\beta \sim N(M, S^2)$
   - $\sigma^2 \sim U(0, C)$

3. **Posterior**: The posterior distribution in this case is $p(\beta, \sigma^2|y, X)$.

- Standardizing the data. It can be a good idea to standardize the data before running MCMC code or even regression in a frequentist setting. Standardizing data means re-scaling the data relative to the mean and standard deviation $z_x = \frac{(x-\mu_x)}{\sigma_x}$, where μ_x is the mean of x and σ_x is the standard deviation of x.

- Recall the t-distribution, this can also be used for a regression setting as the sampling model.

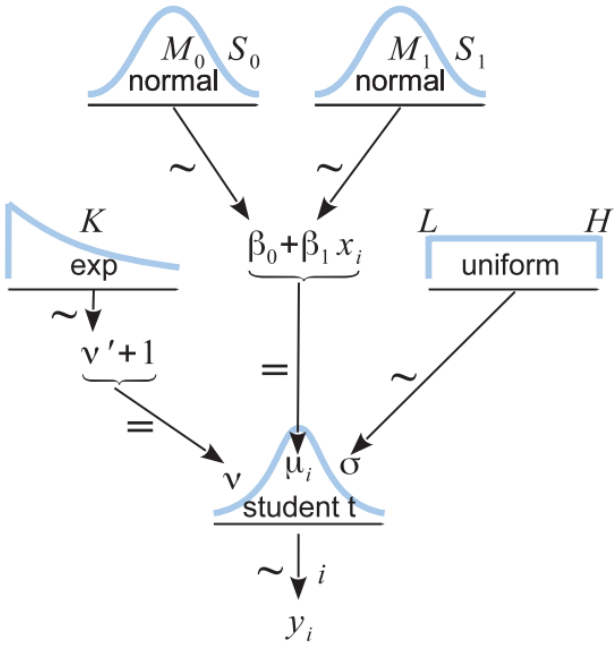


Figure 17.2: A model of dependencies for robust linear regression. The datum, y_i at the bottom of the diagram, is distributed around the central tendency μ_i, which is a linear function of x_i. Compare with Figure 16.11 on p. 437. Copyright © Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2nd Edition.* Academic Press / Elsevier.

JAGS

Let y_i be the total gross earnings of movie i and let x_i be the number of movies released in that month. Then use the sampling model

$$y_i|\beta, \sigma^2 \sim Normal(\beta_0 + x_i\beta_1).$$

All variables are centered and scaled. We use the following priors $\beta_j \sim N(0, 100^2)$ and $\sigma 4 \sim Unif(0, 200, 000)$.

Standardize Variables

```
#Y <- (movie.table$`Total Gross` - mean(movie.table$`Total Gross`)) / sd(movie.table$`Total Gross`)
#x <- (movie.table$Movies - mean(movie.table$Movies)) / sd(movie.table$Movies)
#n <- length(Y)
```

Model Specification

```
model_string <- "model{

# Likelihood
for (i in 1:n){
  Y[i] ~ dnorm(mu[i], 1/sigma^2)
  mu[i] <- beta[1] + beta[2]*x[i]
}

# prior for beta
for (j in 1:2){
  beta[j] ~ dnorm(M,1 / S^2)
}

# Prior for Sigma
sigma ~ dunif(0,C)
}"
```

Compile the Model

```
model <- jags.model(textConnection(model_string), data=list(Y=movie.table$`Total Gross`,
        n=nrow(movie.table), x=movie.table$Movies, M=0, S=100, C=10000), n.adapt = 1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 37
##    Unobserved stochastic nodes: 3
##    Total graph size: 150
##
## Initializing model
```

Draw Samples

```
update(model, 1000)
samples <- coda.samples(model, variable.names = c('beta','sigma'), n.iter = 20000)
summary(samples)
```

```
##
## Iterations = 2001:22000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 20000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean    SD Naive SE Time-series SE
## beta[1]  21.79 55.80 0.394571        1.07674
## beta[2]  14.12  1.38 0.009758        0.02627
## sigma   172.07 21.74 0.153733        0.21368
##
## 2. Quantiles for each variable:
##
##           2.5%    25%    50%    75%  97.5%
## beta[1] -88.74 -15.76  21.85  59.31 130.29
## beta[2]  11.43  13.19  14.12  15.07  16.86
## sigma   135.48 156.85 169.88 184.98 221.06
```

```
#plot(samples)
```

Compare with least squares

```
summary(lm(`Total Gross` ~Movies, data = movie.table))
```

```
##
## Call:
## lm(formula = `Total Gross` ~ Movies, data = movie.table)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -403.00  -91.45  -19.88   61.37  418.41
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   30.574     65.113    0.47    0.642
## Movies        13.929      1.546    9.01 1.21e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 166.5 on 35 degrees of freedom
## Multiple R-squared:  0.6987, Adjusted R-squared:  0.6901
## F-statistic: 81.18 on 1 and 35 DF,  p-value: 1.205e-10
```

Lab Exercises

1. Explore the Seattle Housing dataset graphically and choose a metric variable to use to model housing prices.

```
Seattle <- read.csv('http://math.montana.edu/ahoegh/teaching/stat408/datasets/SeattleHousing.csv',
                    stringsAsFactors = F)
str(Seattle)
```

```
## 'data.frame':    869 obs. of  14 variables:
##  $ price        : num  1350000 228000 289000 720000 247500 ...
##  $ bedrooms     : int  3 3 3 4 3 3 4 5 3 2 ...
##  $ bathrooms    : num  2.5 1 1.75 2.5 1.75 2.5 1 2 2.5 1 ...
##  $ sqft_living  : int  2753 1190 1260 3450 1960 2070 2550 2260 1910 1000 ...
##  $ sqft_lot     : int  65005 9199 8400 39683 15681 13241 4000 12500 66211 10200 ...
##  $ floors       : num  1 1 1 2 1 1.5 2 1 2 1 ...
##  $ waterfront   : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ sqft_above   : int  2165 1190 1260 3450 1960 1270 2370 1130 1910 1000 ...
##  $ sqft_basement: int  588 0 0 0 0 800 180 1130 0 0 ...
##  $ zipcode      : int  98070 98148 98148 98010 98032 98102 98109 98032 98024 98024 ...
##  $ lat          : num  47.4 47.4 47.4 47.3 47.4 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ yr_sold      : int  2015 2014 2014 2015 2015 2014 2014 2014 2015 2014 ...
##  $ mn_sold      : int  3 9 8 3 3 6 6 10 1 11 ...
```

2. Fit a t-distribution regression model for housing price.

Specify the sampling model and all necessary prior distributions.

3. Summarize the results from this model.