Multiple Linear Regression Models

Justin Post

Recap

Given a model, we **fit** the model using data

- Must determine how well the model predicts on **new** data
- Create a test set or use CV
- Judge effectiveness using a **metric** on predictions made from the model

Regression Modeling Ideas

For a set of observations $y_1, \ldots, y_n$, we may want to predict a future value

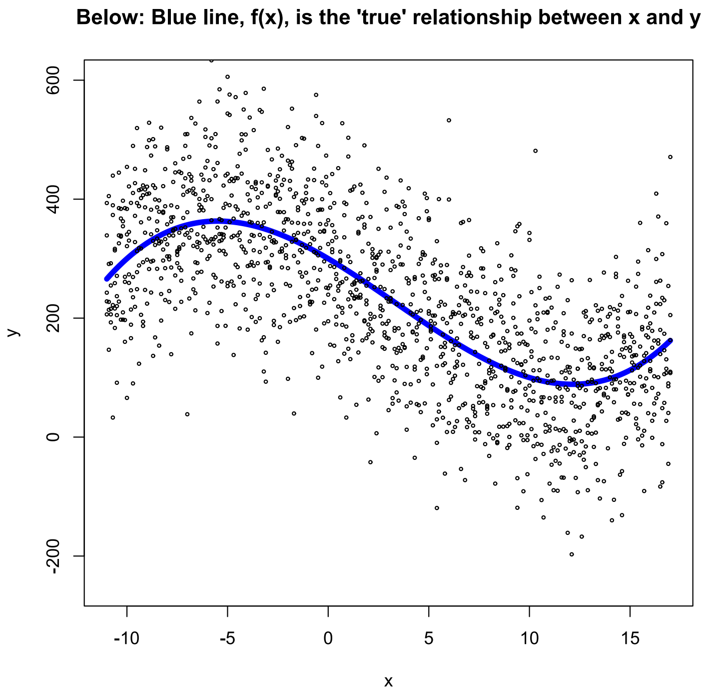
- Often use the sample mean to do so, $\bar{y}$ (an estimate of $E(Y)$)

Regression Modeling Ideas

For a set of observations $y_1, \ldots, y_n$, we may want to predict a future value

- Often use the sample mean to do so, $\bar{y}$ (an estimate of $E(Y)$)

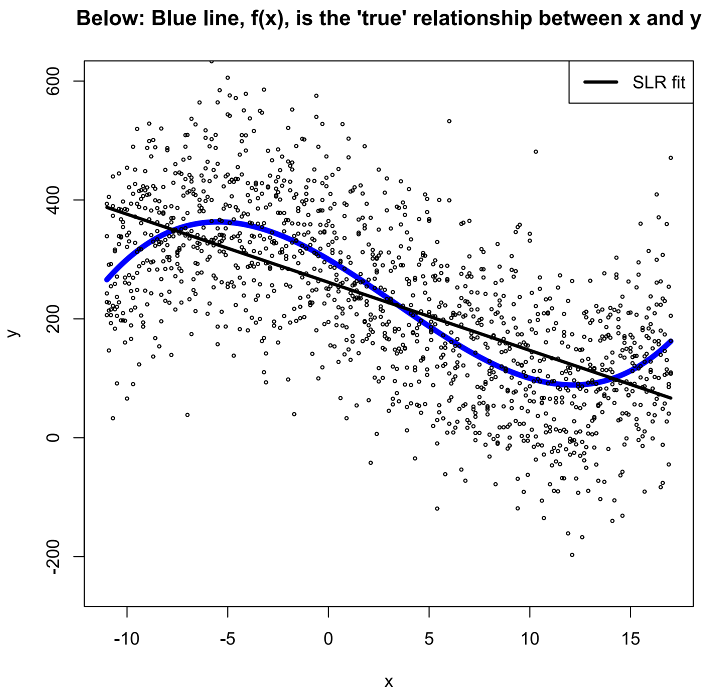
Now consider having pairs $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$



Regression Modeling Ideas

Often use a linear (in the parameters) model for prediction

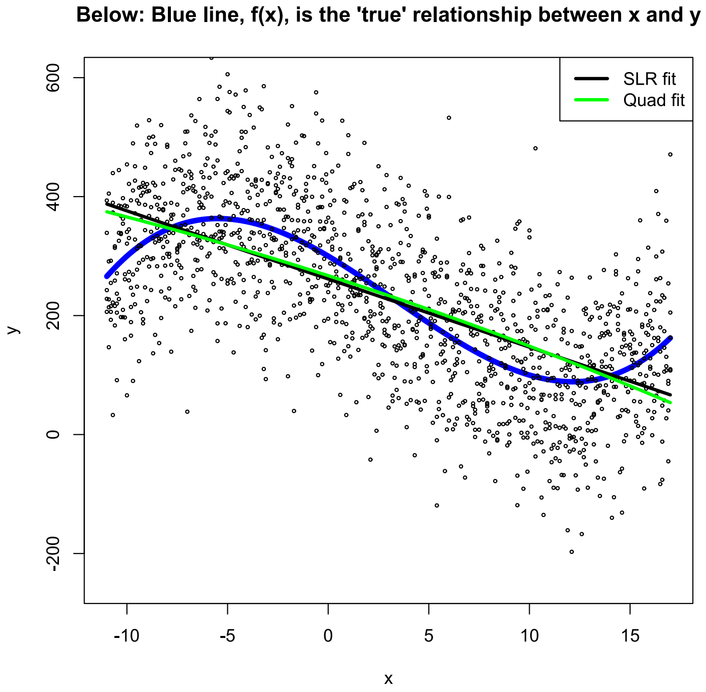
$$\text{SLR model: } E(Y|x) = \beta_0 + \beta_1 x$$



Regression Modeling Ideas

Can include more terms on the right hand side (RHS)

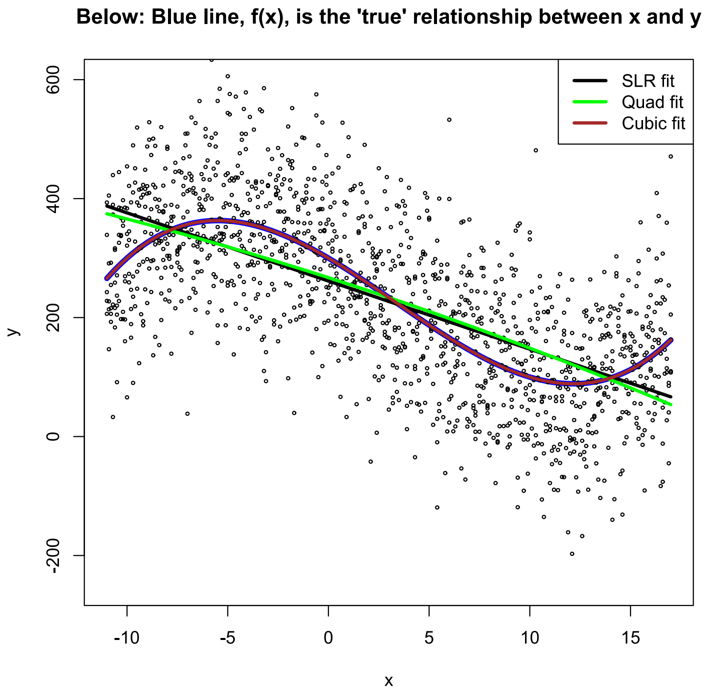
$$\text{Multiple Linear Regression Model: } E(Y|x) = \beta_0 + \beta_1 x + \beta_2 x^2$$



Regression Modeling Ideas

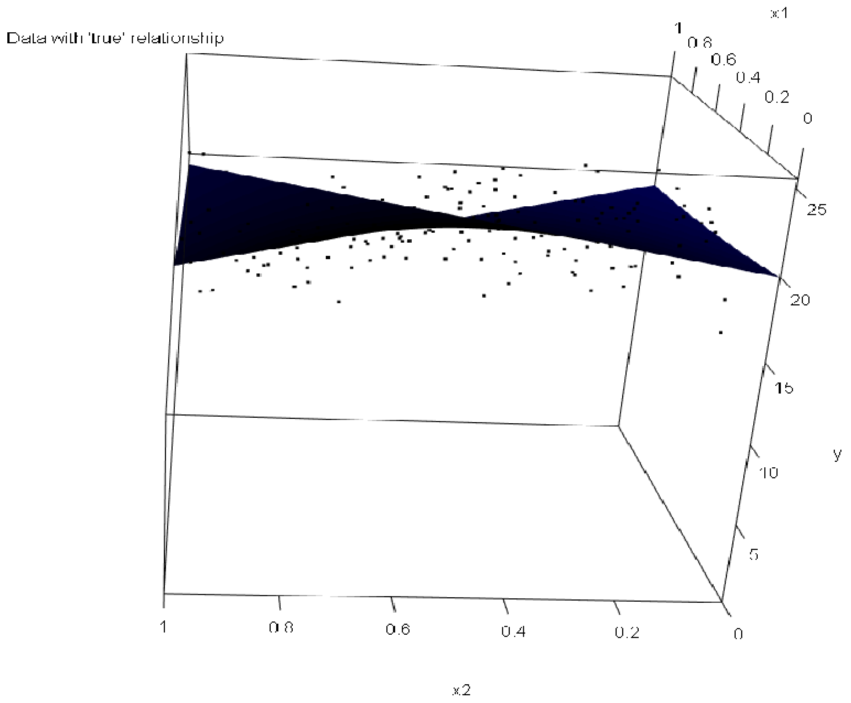
Can include more terms on the right hand side (RHS)

$$\text{Multiple Linear Regression Model: } E(Y|x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$



Regression Modeling Ideas

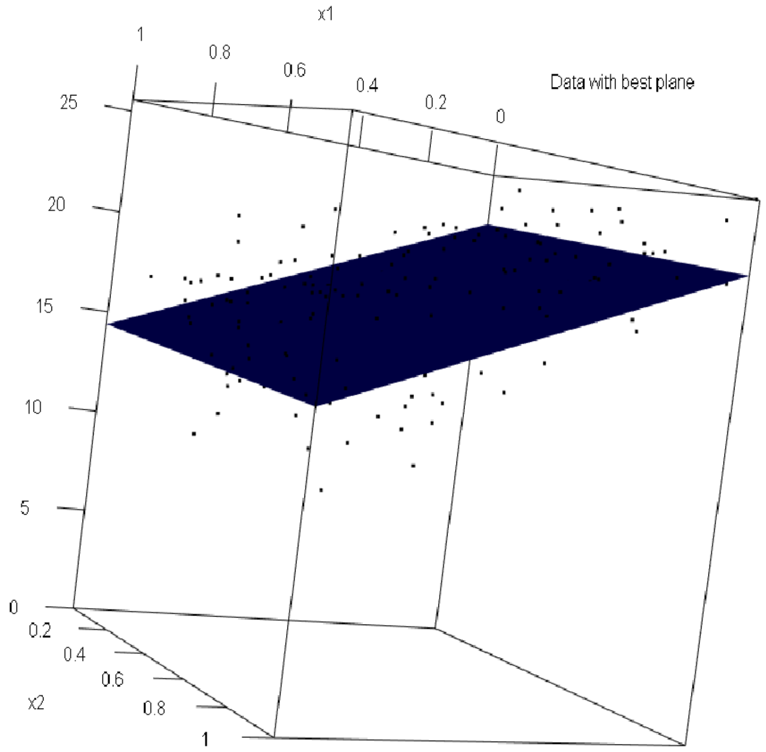
- We model the mean response for a given x value
- With multiple predictors or x's, we do the same idea!



Regression Modeling Ideas

- Including a **main effect** for two predictors fits the best plane through the data

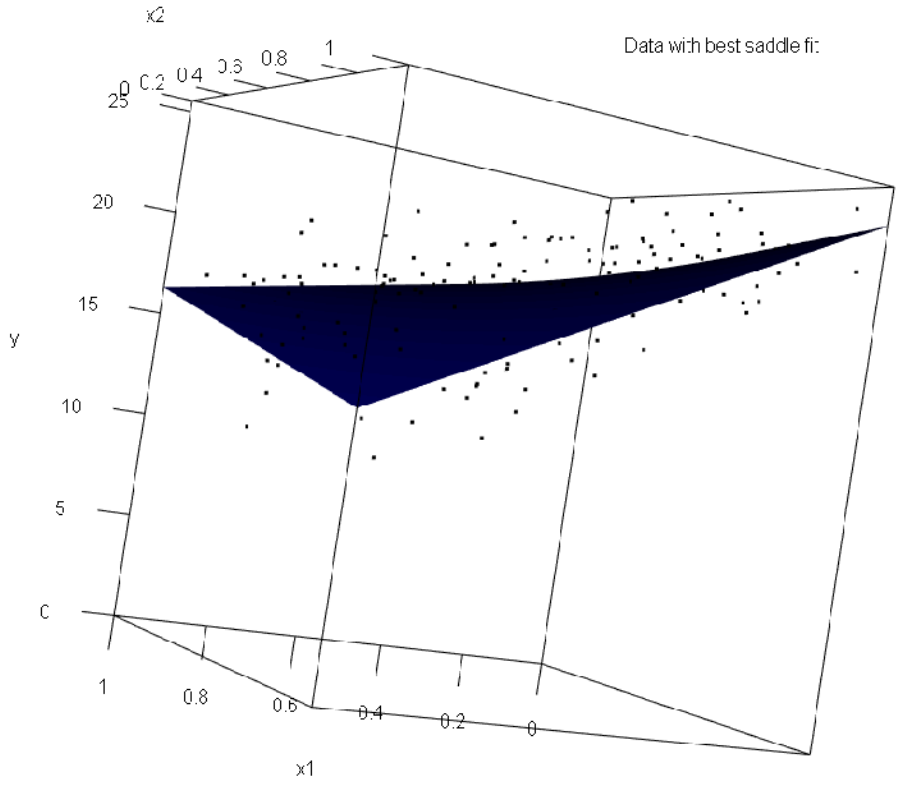
$$\text{Multiple Linear Regression Model: } E(Y|x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$



Regression Modeling Ideas

- Including **main effects** and an **interaction effect** allows for a more flexible surface

Multiple Linear Regression Model: $E(Y|x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$



Regression Modeling Ideas

- Including **main effects** and an **interaction effect** allows for a more flexible surface
- Interaction effects allow for the **effect** of one variable to depend on the value of another
- Model fit previously gives
    - $\hat{y}$ = (19.005) + (-0.791)x1 + (5.631)x2 + (-12.918)x1x2

Regression Modeling Ideas

- Including **main effects** and an **interaction effect** allows for a more flexible surface
- Interaction effects allow for the **effect** of one variable to depend on the value of another
- Model fit previously gives
  - $\hat{y}$ = (19.005) + (-0.791)x1 + (5.631)x2 + (-12.918)x1x2
  - For x_1 = 0, the slope on x_2 is (5.631)`+0*` (-12.918) = 5.631

Regression Modeling Ideas

- Including **main effects** and an **interaction effect** allows for a more flexible surface
- Interaction effects allow for the **effect** of one variable to depend on the value of another
- Model fit previously gives
  - $\hat{y}$ = (19.005) + (-0.791)x1 + (5.631)x2 + (-12.918)x1x2
  - For x_1 = 0, the slope on x_2 is (5.631)`+0*` (-12.918) = 5.631
  - For x_1 = 0.5, the slope on x_2 is (5.631)`+0.5*`(-12.918) = -0.828

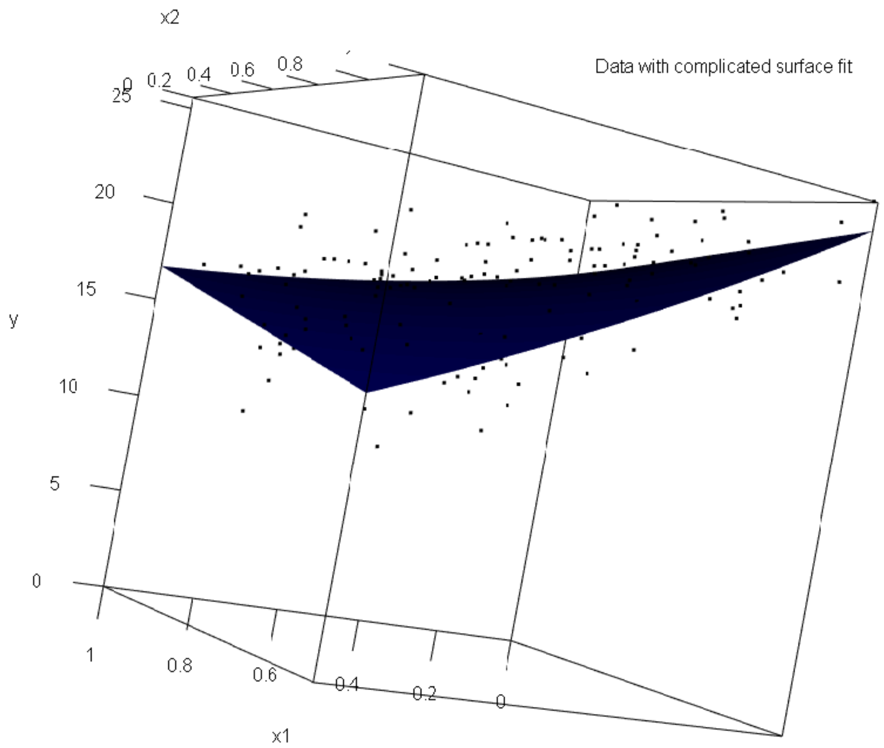
Regression Modeling Ideas

- Including **main effects** and an **interaction effect** allows for a more flexible surface
- Interaction effects allow for the **effect** of one variable to depend on the value of another
- Model fit previously gives
  - $\hat{y}$ = (19.005) + (-0.791)x1 + (5.631)x2 + (-12.918)x1x2
  - For x_1 = 0, the slope on x_2 is (5.631)`+0*` (-12.918) = 5.631
  - For x_1 = 0.5, the slope on x_2 is (5.631)`+0.5*`(-12.918) = -0.828
  - For x_1 = 1, the slope on x_2 is (5.631)`+1*`(-12.918) = -7.286
- Similarly, the slope on x_1 depends on x_2!

Regression Modeling Ideas

- Including **main effects** and an **interaction effect** allows for a more flexible surface
- Can also include higher order polynomial terms

Multiple Linear Regression Model: $E(Y|x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2$



Regression Modeling Ideas

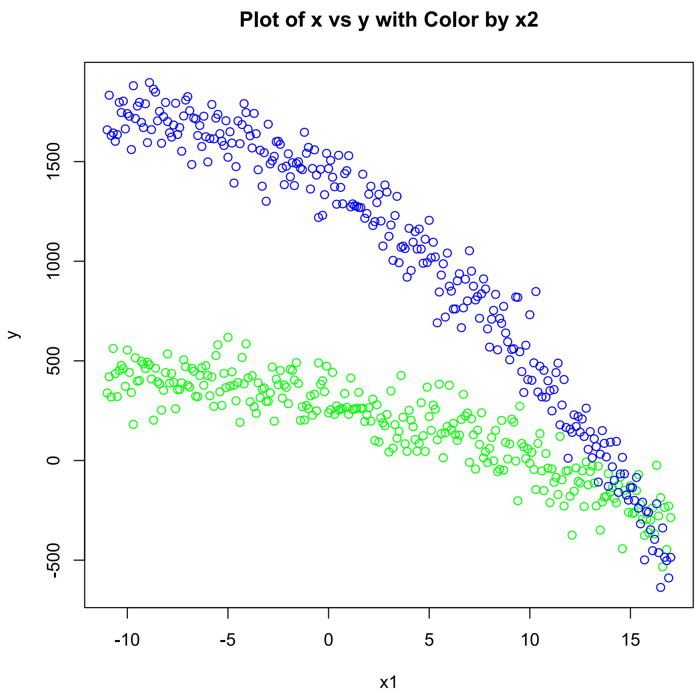
Can also include categorical variables through **dummy** or **indicator** variables

- Categorical variable with value of $Success$ and $Failure$
- Define $x_2 = 0$ if variable is $Failure$
- Define $x_2 = 1$ if variable is $Success$

Regression Modeling Ideas

Can also include categorical variables through **dummy** or **indicator** variables

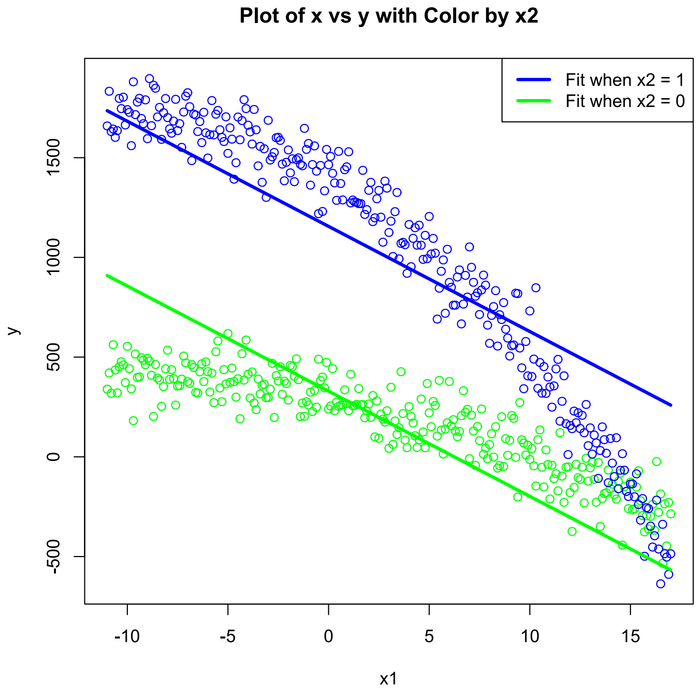
- Categorical variable with value of $Success$ and $Failure$
- Define $x_2 = 0$ if variable is $Failure$
- Define $x_2 = 1$ if variable is $Success$



Regression Modeling Ideas

- Define $x_2 = 0$ if variable is $Failure$
- Define $x_2 = 1$ if variable is $Success$

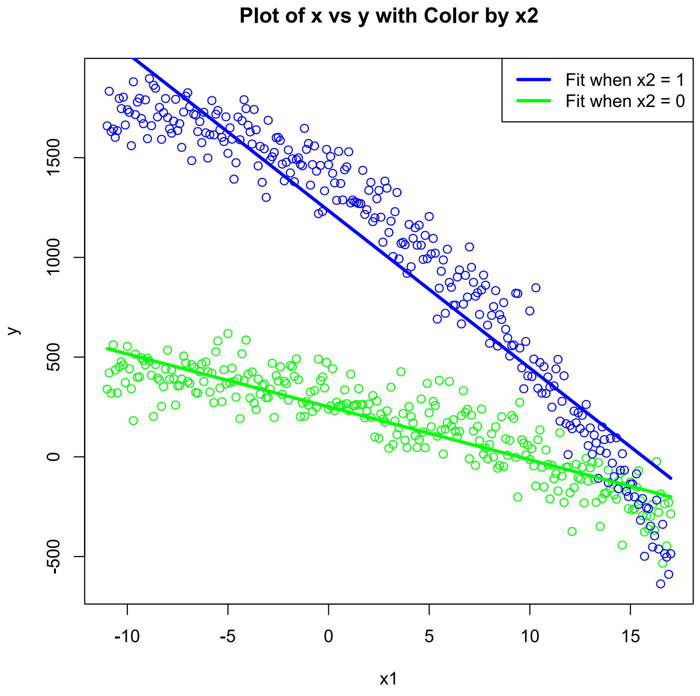
$$\text{Separate Intercept Model: } E(Y|x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$



Regression Modeling Ideas

- Define $x_2 = 0$ if variable is $Failure$
- Define $x_2 = 1$ if variable is $Success$

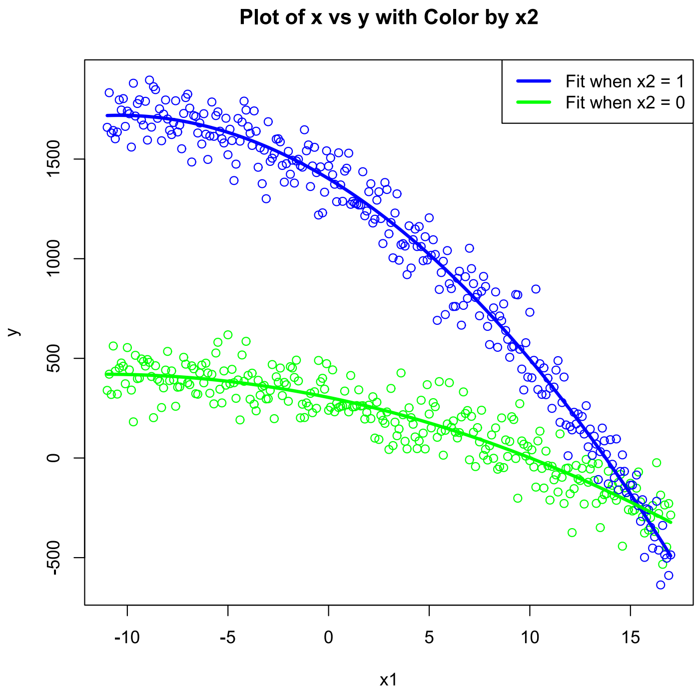
$$\text{Separate Intercept and Slopes Model: } E(Y|x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$$



Regression Modeling Ideas

- Define $x_2 = 0$ if variable is $Failure$
- Define $x_2 = 1$ if variable is $Success$

Separate Quadratics Model: $E(Y|x) = \beta_0 + \beta_1 x_2 + \beta_2 x_1 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_1^2 x_2$



Regression Modeling Ideas

If your categorical variable has more than k>2 categories, define k-1 dummy variables

- Categorical variable with values of "Assistant", "Contractor", "Executive"
- Define $x_2 = 0$ if variable is $Executive$ or $Contractor$
- Define $x_2 = 1$ if variable is $Assistant$
- Define $x_3 = 0$ if variable is $Contractor$ or $Assistant$
- Define $x_3 = 1$ if variable is $Executive$

Regression Modeling Ideas

If your categorical variable has more than k>2 categories, define k-1 dummy variables

- Categorical variable with values of "Assistant", "Contractor", "Executive"
- Define $x_2 = 0$ if variable is $Executive$ or $Contractor$
- Define $x_2 = 1$ if variable is $Assistant$
- Define $x_3 = 0$ if variable is $Contractor$ or $Assistant$
- Define $x_3 = 1$ if variable is $Executive$

$$\text{Separate Intercepts Model: } E(Y|x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

What is implied if x_2 and x_3 are both zero?

Fitting an MLR Model

Big Idea: Trying to find the line, plane, saddle, etc. **of best fit** through points

- How do we do the fit??
    - Usually minimize the sum of squared residuals (errors)

Fitting an MLR Model

Big Idea: Trying to find the line, plane, saddle, etc. **of best fit** through points

- How do we do the fit??
  - Usually minimize the sum of squared residuals (errors)
- Residual = observed - predicted or $y_i - \hat{y}_i$

$$\min_{\hat{\beta}'s} \sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \ldots + \hat{\beta}_p x_{pi}))^2$$

Fitting an MLR Model

Big Idea: Trying to find the line, plane, saddle, etc. **of best fit** through points

- How do we do the fit??
  - Usually minimize the sum of squared residuals (errors)
- Residual = observed - predicted or $y_i - \hat{y}_i$

$$\min_{\hat{\beta}'s} \sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \ldots + \hat{\beta}_p x_{pi}))^2$$

- Closed-form results exist for easy calculation via software!

Fitting a Linear Regression Model in R

- Use `lm()` and specify a `formula`: `LHS ~ RHS`
  - `y ~` implies y is modelled by a linear function of the RHS
  - RHS consists of terms separated by `+` operators
    - `y ~ x1 + x2` gives $E(Y|x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

Fitting a Linear Regression Model in R

- Use `lm()` and specify a `formula`: `LHS ~ RHS`
  - `y ~` implies y is modelled by a linear function of the RHS
  - RHS consists of terms separated by `+` operators
    - `y ~ x1 + x2` gives $E(Y|x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
    - `:` for interactions, `y ~ x1 + x2 + x1:x2` gives $E(Y|x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$

Fitting a Linear Regression Model in R

- Use `lm()` and specify a `formula`: `LHS ~ RHS`
  - `y ~` implies y is modelled by a linear function of the RHS
  - RHS consists of terms separated by `+` operators
    - `y ~ x1 + x2` gives $E(Y|x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
    - `:` for interactions, `y ~ x1 + x2 + x1*x2` gives $E(Y|x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$
    - `*` denotes factor crossing: `a*b` is interpreted as `a + b + a:b`
    - `y ~ x - 1` removes the intercept term

Fitting a Linear Regression Model in R

- Use `lm()` and specify a `formula`: `LHS ~ RHS`
  - `y ~` implies y is modelled by a linear function of the RHS
  - RHS consists of terms separated by `+` operators
    - `y ~ x1 + x2` gives $E(Y|x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
    - `:` for interactions, `y ~ x1 + x2 + x1*x2` gives $E(Y|x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$
    - `*` denotes factor crossing: `a*b` is interpreted as `a + b + a:b`
    - `y ~ x - 1` removes the intercept term
    - `I()` can be used to create arithmetic predictors
      - `y ~ a + I(b+c)` implies `b+c` is the sum of b and c
      - `y ~ x + I(x^2)` implies $E(Y|x_1) = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2$

Fitting MLR Models

- Let's read in our `bike_data` and fit some MLR models

```
library(tidyverse)
bike_data <- read_csv("https://www4.stat.ncsu.edu/~online/datasets/bikeDetails.csv")
bike_data <- bike_data |>
  mutate(log_selling_price = log(selling_price),
         log_km_driven = log(km_driven)) |>
  select(log_km_driven, year, log_selling_price, owner, everything())
bike_data
```

```
## # A tibble: 1,061 × 9
##   log_km_driven  year log_selling_price owner    name  selling_price seller_type
##           <dbl> <dbl>             <dbl> <chr>    <chr>         <dbl> <chr>
## 1          5.86  2019              12.1 1st own… Roya…        175000 Individual
## 2          8.64  2017              10.7 1st own… Hond…         45000 Individual
## 3          9.39  2018              11.9 1st own… Roya…        150000 Individual
## 4         10.0   2015              11.1 1st own… Yama…         65000 Individual
## 5          9.95  2011               9.90 2nd own… Yama…         20000 Individual
## # ℹ 1,056 more rows
## # ℹ 2 more variables: km_driven <dbl>, ex_showroom_price <dbl>
```

Fitting MLR Models

- Create models with the same slope but intercepts differing by a categorical variable

```r
owner_fits <- lm(log_selling_price ~ owner + log_km_driven, data = bike_data)
coef(owner_fits)
```

```
##    (Intercept) owner2nd owner owner3rd owner owner4th owner  log_km_driven
##    14.62423775    -0.06775874     0.08148045     0.20110313    -0.38930862
```

Fitting MLR Models

- Create a data frame for plotting

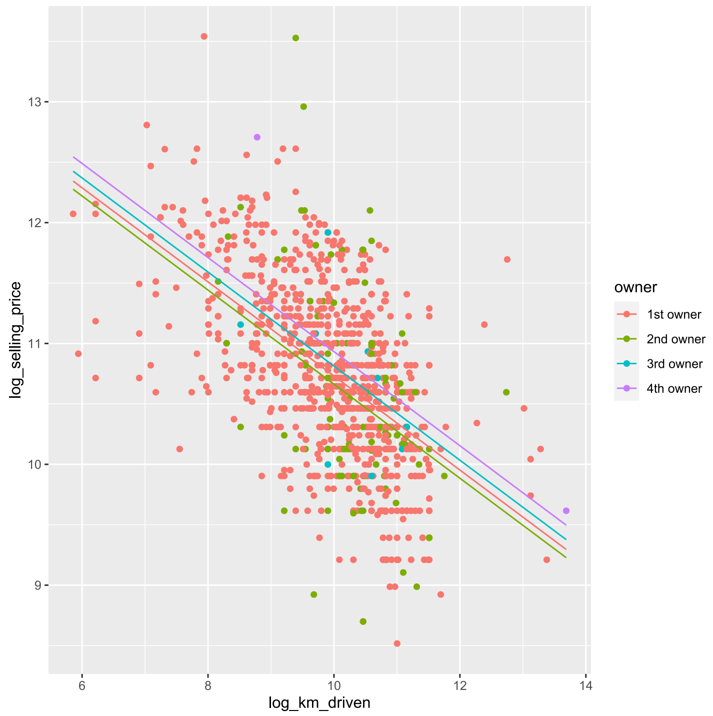
```
x_values <- seq(from = min(bike_data$log_km_driven),
                to = max(bike_data$log_km_driven),
                length = 2)
pred_df <- data.frame(log_km_driven = rep(x_values, 4),
                      owner = c(rep("1st owner", 2),
                                rep("2nd owner", 2),
                                rep("3rd owner", 2),
                                rep("4th owner", 2)))
pred_df <- pred_df |>
  mutate(predictions = predict(owner_fits, newdata = pred_df))
pred_df
```

```
##   log_km_driven     owner predictions
## 1      5.857933 1st owner   12.343694
## 2     13.687677 1st owner    9.295507
## 3      5.857933 2nd owner   12.275935
## 4     13.687677 2nd owner    9.227748
## 5      5.857933 3rd owner   12.425174
## 6     13.687677 3rd owner    9.376987
## 7      5.857933 4th owner   12.544797
## 8     13.687677 4th owner    9.496610
```

Fitting MLR Models

- Plot our different intercept models

```
ggplot(bike_data, aes(x = log_km_driven, y = log_selling_price, color = owner)) +
  geom_point() +
  geom_line(data = pred_df, aes(x = log_km_driven, y = predictions, color = owner))
```



Fitting MLR Models

- Create models with the different slopes and intercepts

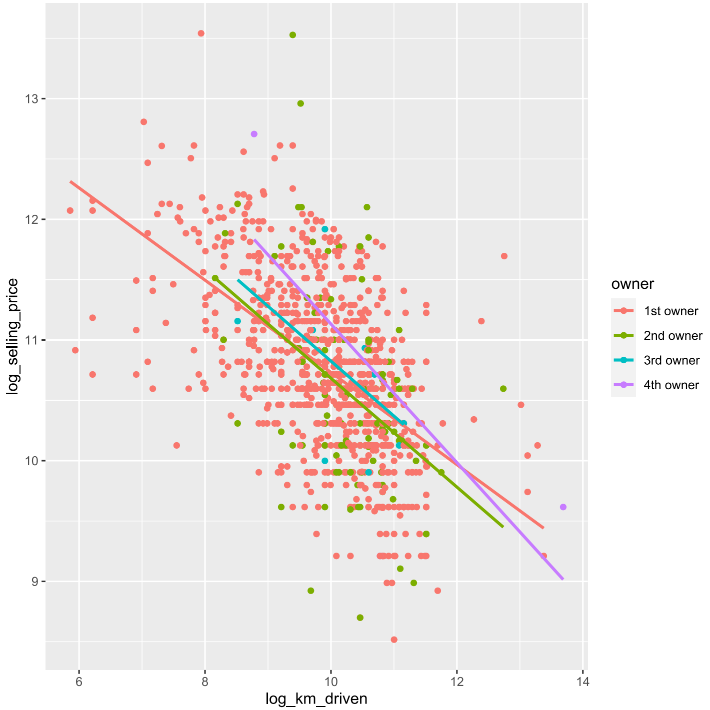
```
owner_fits_full <- lm(log_selling_price ~ owner*log_km_driven, data = bike_data)
coef(owner_fits_full)
```

```
##                  (Intercept)                owner2nd owner 
##                  14.55347484                    0.63862406 
##               owner3rd owner                owner4th owner 
##                   0.82280649                    2.31991467 
##                log_km_driven owner2nd owner:log_km_driven 
##                  -0.38219492                   -0.06871037 
## owner3rd owner:log_km_driven owner4th owner:log_km_driven 
##                  -0.07295150                   -0.19192122 
```

Fitting MLR Models

- Plot our different intercept models

```r
ggplot(bike_data, aes(x = log_km_driven, y = log_selling_price, color = owner)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```



Choosing an MLR Model

- Given a bunch of predictors, tons of models you could fit! How to choose?
- Many variable selection methods exist...
- If you care mainly about prediction, just use *cross-validation* or training/test split!
  - Compare predictions using some metric!
  - We'll see how to use `tidymodels` to do this in a coherent way shortly!

Recap

- Multiple Linear Regression models are a common model used for a numeric response
- Generally fit via minimizing the sum of squared residuals or errors
    - Could fit using sum of absolute deviation, or other metric
- Can include polynomial terms, interaction terms, and categorical variables
- Good metric to compare models with a continuous response is the RMSE