BUS41100 Applied Regression Analysis

Week 3: Multiple Linear Regression

Polynomials, log transformation,
categorical variables, interactions & main effects

Max H. Farrell
The University of Chicago Booth School of Business

Multiple vs simple linear regression

Fundamental model is the same.

Basic concepts and techniques translate directly from SLR.

- Individual parameter inference and estimation is the same, conditional on the rest of the model.
- We still use `lm`, `summary`, `predict`, etc.

The hardest part would be moving to matrix algebra to translate all of our equations. Luckily, R does all that for you.

Polynomial regression

A nice bridge between SLR and MLR is polynomial regression.

Still only one X variable, but we add powers of X:

$$\mathbb{E}[Y|X] = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_m X^m$$

You can fit any mean function if m is big enough.

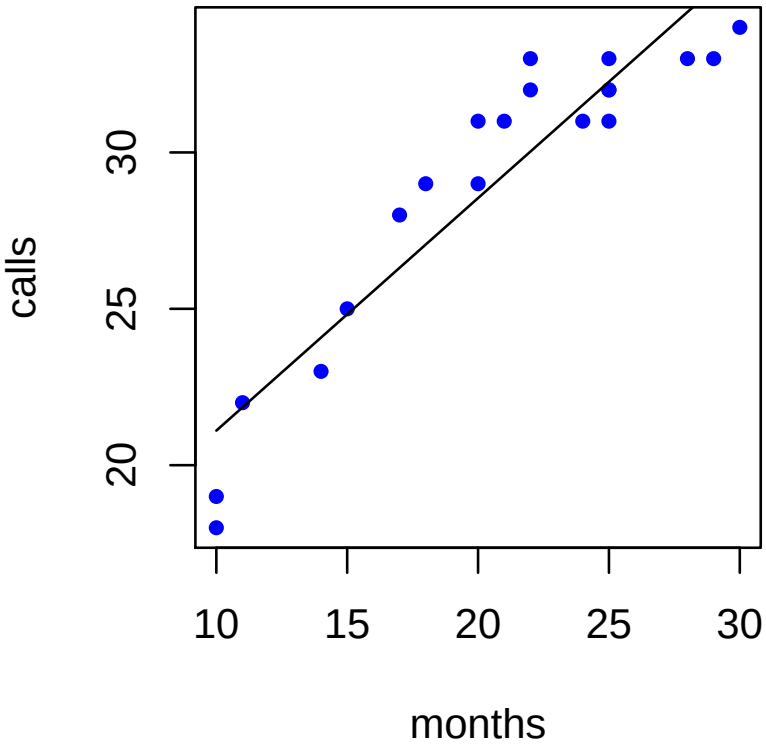
- Usually, $m = 2$ does the trick.

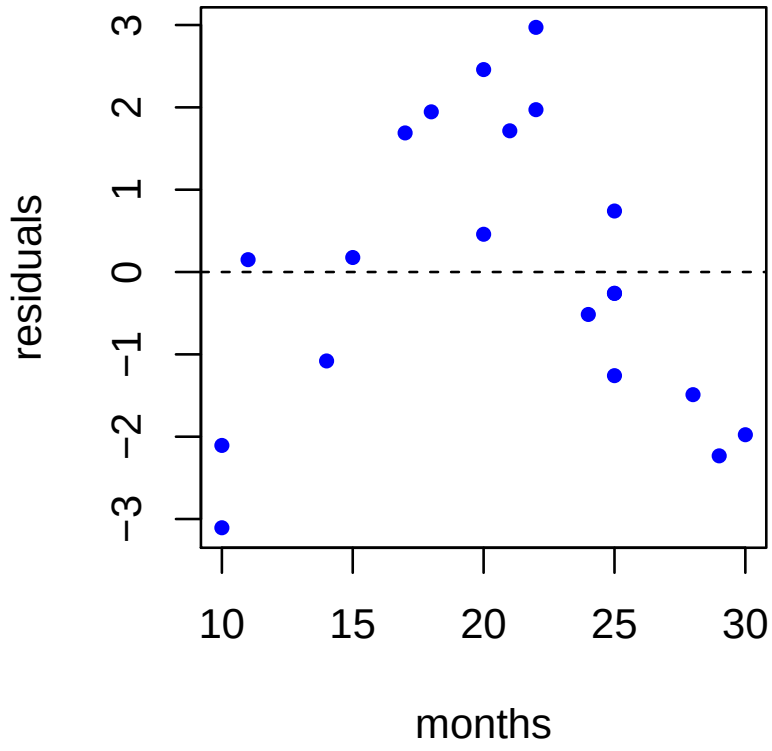
This is our first "*multiple* linear regression"!

Example: telemarketing/call-center data.

- How does length of employment (`months`) relate to productivity (number of `calls` placed per day)?

```
> attach(telemkt <- read.csv("telemarketing.csv"))
> tele1 <- lm(calls~months)
> xgrid <- data.frame(months = 10:30)
> par(mfrow=c(1,2))
> plot(months, calls, pch=20, col=4)
> lines(xgrid$months, predict(tele1, newdata=xgrid))
> plot(months, tele1$residuals, pch=20, col=4)
> abline(h=0, lty=2)
```





It looks like there is a polynomial shape to the residuals.

- We are leaving some predictability on the table
  . . . just not linear predictability.

Testing for nonlinearity

To see if you need more nonlinearity, try the regression which includes the next polynomial term, and see if it is significant.

For example, to see if you need a quadratic term,

- fit the model then run the regression $\mathbb{E}[Y|X] = \beta_0 + \beta_1 X + \beta_2 X^2$.
- If your test implies $\beta_2 \neq 0$, you need X^2 in your model.

Note: p-values are calculated "given the other β's are nonzero"; i.e., conditional on X being in the model.

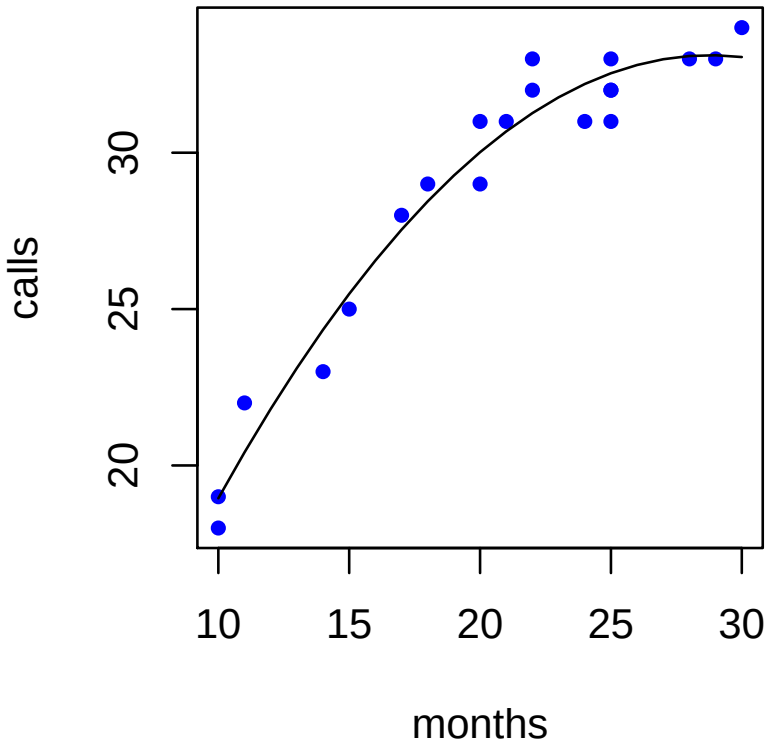
Test for a quadratic term:

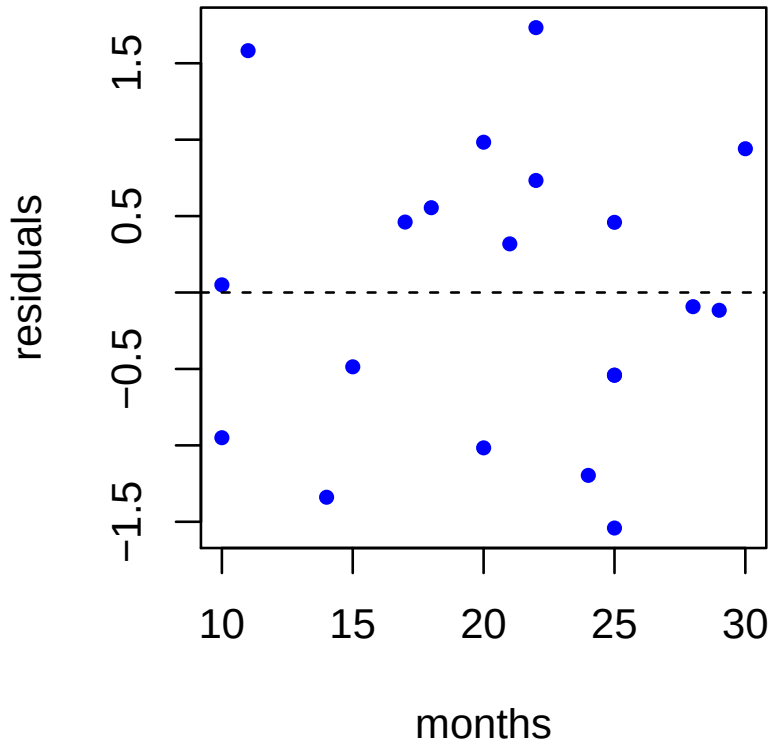
```
> months2 <- months^2
> tele2 <- lm(calls~ months + months2)
> summary(tele2) ## abbreviated output

Coefficients:
             Estimate  Std. Err  t value Pr(>|t|)
(Intercept) -0.140471   2.32263  -0.060    0.952
months       2.310202   0.25012   9.236 4.90e-08 ***
months2     -0.040118   0.00633  -6.335 7.47e-06 ***
```

The quadratic $\texttt{months}^2$ term has a very significant t-value, so a better model is $\texttt{calls} = \beta_0 + \beta_1\texttt{months} + \beta_2\texttt{months}^2 + \varepsilon$.

Everything looks much better with the quadratic mean model.

```
> xgrid <- data.frame(months=10:30, months2=(10:30)^2)
> par(mfrow=c(1,2))
> plot(months, calls, pch=20, col=4)
> lines(xgrid$months, predict(tele2, newdata=xgrid))
> plot(months, tele2$residuals, pch=20, col=4)
> abline(h=0, lty=2)
```





A few words of caution

We can always add higher powers (cubic, etc.) if necessary.

- If you add a higher order term, the lower order term is kept *regardless* of its individual t-stat.

(see handout on website)

Be very careful about predicting outside the data range;

- the curve may do unintended things beyond the data.

Watch out for over-fitting.

- You can get a "perfect" fit with enough polynomial terms,
- but that doesn't mean it will be any good for prediction or understanding.

The log-log model

The other common covariate transform is $\log(X)$.

- When X-values are bunched up, $\log(X)$ helps spread them out and reduces the leverage of extreme values.
- Recall that both reduce s_{b_1}.

In practice, this is often used in conjunction with a $\log(Y)$ response transformation.

- The log-log model is

$$\log(Y) = \beta_0 + \beta_1 \log(X) + \varepsilon.$$

- It is super useful, and has some special properties ...

Recall that

- $\log$ is always natural log, with base $e = 2.718\ldots$, and
- $\log(ab) = \log(a) + \log(b)$
- $\log(a^b) = b\log(a)$.

Consider the multiplicative model $\mathbb{E}[Y|X] = AX^B$.
Take logs of both sides to get

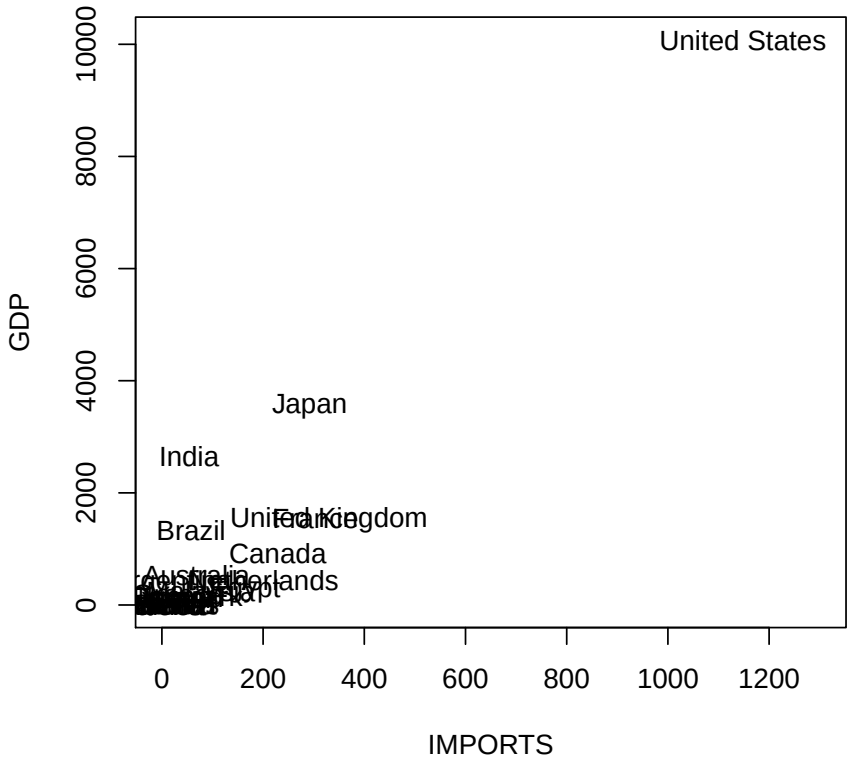
$$\log(\mathbb{E}[Y|X]) = \log(A) + \log(X^B) = \log(A) + B\log(X)$$
$$\equiv \beta_0 + \beta_1 \log(X).$$

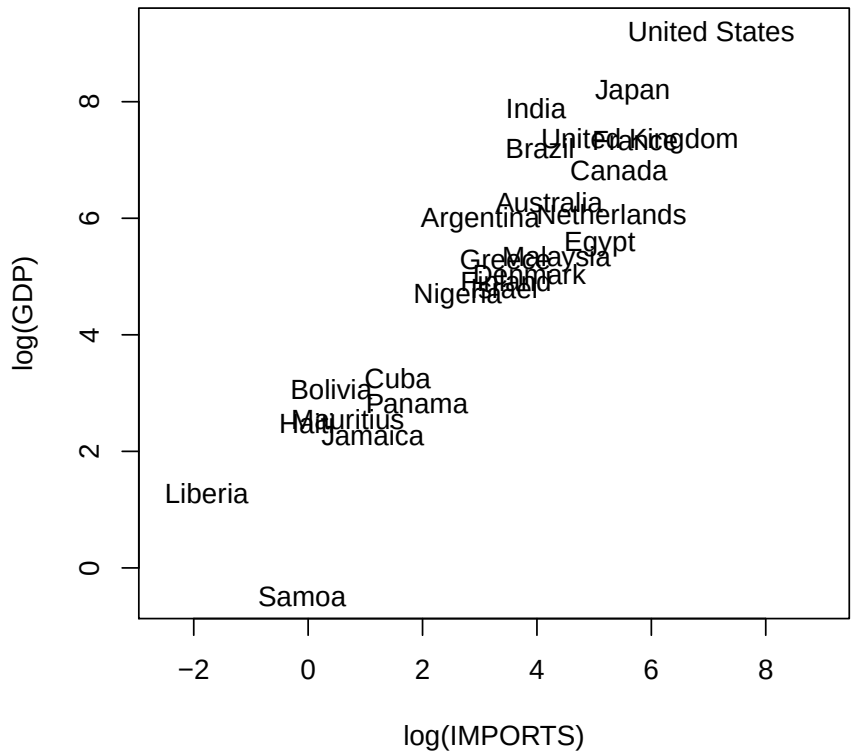
The log-log model is appropriate whenever things are linearly related on a multiplicative, or percentage, scale.

(See handout on course website.)

Consider a country's GDP as a function of IMPORTS:

- Since trade multiplies, we might expect to see %GDP increase with %IMPORTS.





Elasticity and the log-log model

In a log-log model, the slope β_1 is sometimes called elasticity.

An elasticity is (roughly) % change in Y per 1% change in X.

$$\beta_1 \approx \frac{d\%Y}{d\%X}$$

For example, economists often assume that GDP has import elasticity of 1. Indeed:

```
> coef(lm(log(GDP) ~ log(IMPORTS)))
 (Intercept)  log(IMPORTS)
      1.8915         0.9693
```

(Can we test for 1%?)

Price elasticity

In marketing, the slope coefficient β_1 in the regression

$$\log(\text{sales}) = \beta_0 + \beta_1 \log(\text{price}) + \varepsilon$$

is called price elasticity:

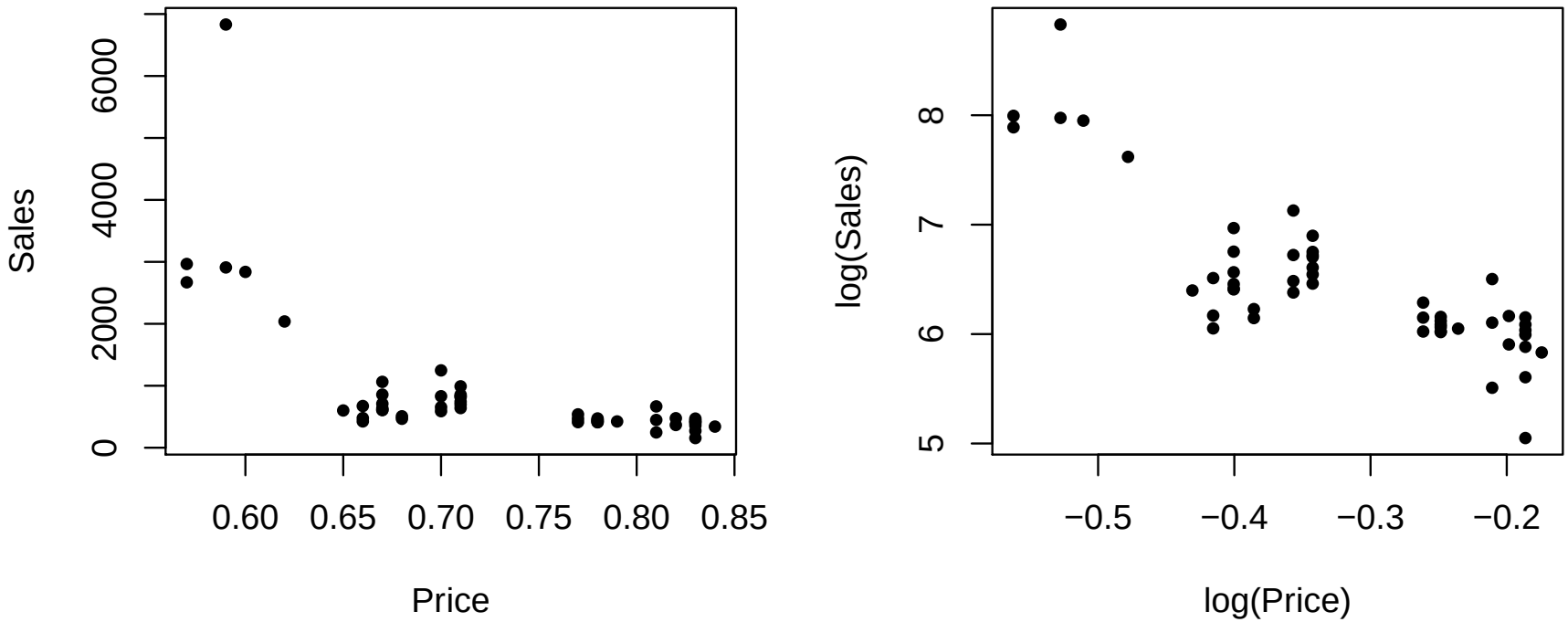
- the % change in `sales` per 1% change in `price`.

The model implies that $\mathbb{E}[\text{sales}|\text{price}] = A * \text{price}^{\beta_1}$ such that β_1 is the constant rate of change.

Economists have "demand elasticity" curves, which are just more general and harder to measure.

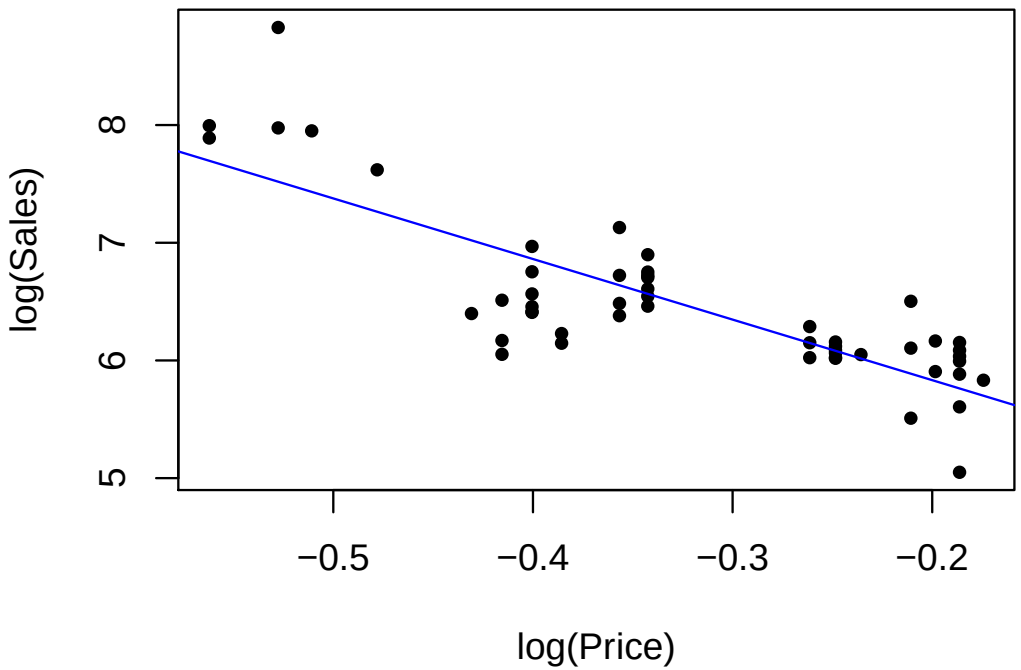
Example: we have Nielson SCANTRACK data on supermarket sales of a canned food brand produced by Consolidated Foods.

```
> attach(confood <- read.csv("confood.csv"))
> par(mfrow=c(1,2))
> plot(Price,Sales, pch=20)
> plot(log(Price),log(Sales), pch=20)
```



Run the regression to determine price elasticity:

```
> confood.reg <- lm(log(Sales) ~ log(Price))
> coef(confood.reg)
(Intercept)  log(Price)
   4.802877   -5.147687
> plot(log(Price),log(Sales), pch=20)
> abline(confood.reg, col=4)
```



- Sales decrease by about 5% for every 1% price increase.

Beyond SLR

Many problems involve more than one independent variable or factor which affects the dependent or response variable.

- Multi-factor asset pricing models (beyond CAPM).
- Demand for a product given prices of competing brands, advertising, household attributes, etc.
- More than size to predict house price!

In SLR, the conditional mean of Y depends on X. The multiple linear regression (MLR) model extends this idea to include more than one independent variable.

The MLR Model

The MLR model is same as always, but with more covariates.

$$Y|X_1, \ldots, X_d \sim \mathcal{N}(\beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d, \sigma^2)$$

Recall the key assumptions of our linear regression model:

(i) The conditional mean of Y is linear in the X_j variables.

(ii) The additive errors (deviations from line)

- are Normally distributed
- independent from each other and all the X_j
- identically distributed (i.e., they have constant variance)

Our interpretation of regression coefficients can be extended from the simple single covariate regression case:

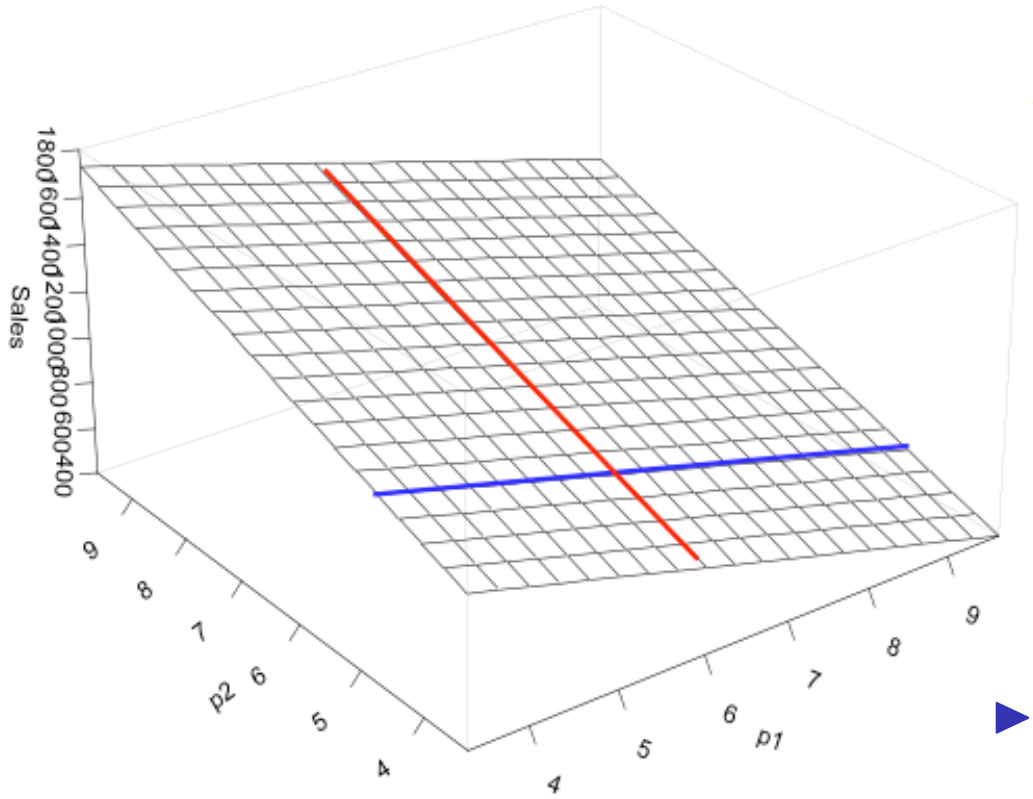
$$\beta_j = \frac{\partial \mathbb{E}[Y|X_1, \ldots, X_d]}{\partial X_j}$$

- Holding all other variables constant, β_j is the average change in Y per unit change in X_j.

∂ is from calculus and means "change in"

If $d = 2$, we can plot the regression surface in 3D.

Consider `sales` of a product as predicted by price of this product (P1) and the price of a competing product (P2).



Sales = 1 – 1.0*P1 + 1.1*P2

hold P2 fixed and vary P1

hold P1 fixed and vary P2

- Everything on log scale

  $\Rightarrow$ -1.0 & 1.1 are elasticities

Obtaining these estimates in R is very easy:

```
> salesdata <- read.csv("sales.csv")
> attach(salesdata)
> salesMLR <- lm(Sales ~ P1 + P2)
> salesMLR

Call:
lm(formula = Sales ~ P1 + P2)

Coefficients:
(Intercept)           P1           P2
      1.003       -1.006        1.098
```

Same Least Squares Principles

How do we estimate the MLR model parameters?

- fitted values $\hat{Y}_i = b_0 + b_1 X_{1i} + b_2 X_{2i} + \cdots + b_d X_{di}$
- residuals $e_i = Y_i - \hat{Y}_i$
- residual variance $s^2 = \frac{1}{n-p} \sum_{i=1}^{n} e_i^2$, $p = d + 1$

Then find the best fitting plane, i.e., coefs $b_0, b_1, b_2, \ldots, b_d$, by minimizing the sum of squared errors, s^2.

Residuals in MLR

As in the SLR model, the residuals in multiple regression are purged of any linear relationship to the independent variables.

We decompose Y into the part predicted by X and the part due to idiosyncratic error.

$$Y = \hat{Y} + e$$

$$\text{corr}(X_j, e) = 0 \qquad \text{corr}(\hat{Y}, e) = 0$$

These hold by construction, just like SLR.

Inference for coefficients

As before in SLR, the LS linear coefficients are random (different for each sample) and correlated with each other.

The sampling distribution for the whole vector $\mathbf{b} = [b_0, b_1, \cdots, b_d]$ is a multivariate Normal:

$$\mathbf{b} \sim \mathcal{N}_{d+1}(\boldsymbol{\beta}, \boldsymbol{\Sigma}_{\mathbf{b}})$$

- With mean $\boldsymbol{\beta} = [\beta_0, \beta_1, \cdots, \beta_d]'$ (unbiased, as before)
- Variance-covariance matrix $\boldsymbol{\Sigma}_{\mathbf{b}}$
- Same as last week:

$$\begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \sim \mathcal{N}_2 \left(\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \begin{bmatrix} \sigma^2_{b_0} & \mathrm{cov}(b_0, b_1) \\ \mathrm{cov}(b_0, b_1) & \sigma^2_{b_1} \end{bmatrix} \right)$$

Coefficient covariance matrix

$\boldsymbol{\Sigma}_{\mathbf{b}} = \text{var}(\mathbf{b})$: the $p \times p$ covariance matrix for random vector $\mathbf{b}$

$$\boldsymbol{\Sigma}_{\mathbf{b}} = \begin{bmatrix} \text{var}(b_0) & \text{cov}(b_0, b_1) & & \\ \text{cov}(b_1, b_0) & \text{var}(b_1) & & \\ & & \ddots & \\ & & \text{var}(b_{d-1}) & \text{cov}(b_{d-1}, b_d) \\ & & \text{cov}(b_d, b_{d-1}) & \text{var}(b_d) \end{bmatrix}$$

- Variance decreases with n and $\text{var}(\mathbf{X})$; increases with σ^2.

$\Rightarrow$ Standard errors are the square root of the diagonal of $\boldsymbol{\Sigma}_{\mathbf{b}}$.

Inference for individual coefficients

Intervals and t-statistics are exactly the same as in SLR.

- A $(1-\alpha)100\%$ C.I. for β_j is $b_j \pm z_{\alpha/2} s_{b_j}$.
- $z_{b_j} = (b_j - \beta_j^0)/s_{b_j} \sim \mathcal{N}(0,1)$ is number of standard errors between the LS estimate and the null value.

Intervals/testing via b_j & s_{b_j} are one-at-a-time procedures:

- You are evaluating the j^{th} coefficient conditional on the other X's being in the model, but regardless of the values you've estimated for the other b's.

Conveniently, R's `summary` gives you all the standard errors.

(or do it manually, see `week3-Rcode.R`)

```
> summary(salesMLR) ## abbreviated output

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.002688   0.007445   134.7   <2e-16 ***
P1          -1.005900   0.009385  -107.2   <2e-16 ***
P2           1.097872   0.006425   170.9   <2e-16 ***

Residual standard error: 0.01453 on 97 degrees of freedom
Multiple R-squared:  0.998, Adjusted R-squared:  0.9979
F-statistic: 2.392e+04 on 2 and 97 DF,  p-value: < 2.2e-16
```

Forecasting in MLR

Prediction follows exactly the same methodology as in SLR.

For new data $\mathbf{x}_f = [X_{1,f} \cdots X_{d,f}]'$,

- $\hat{Y}_f = b_0 + b_1 X_{1f} + \cdots + b_d X_{df}$
- $\text{var}[Y_f|\mathbf{x}_f] = \text{var}(\hat{Y}_f) + \text{var}(\varepsilon_f) = s^2_{\text{fit}} + s^2 = s^2_{\text{pred}}$.
- $(1-\alpha)$ level prediction interval is still $\hat{Y}_f \pm z_{\alpha/2} s_{\text{pred}}$.

The syntax in R is also exactly the same as before:

```
> predict(salesMLR, data.frame(P1=1, P2=1),
+   interval="prediction", level=0.95)
       fit      lwr      upr
1 1.094661 1.064015 1.125306

> predict(salesMLR, data.frame(P1=1, P2=1),
+   se.fit=TRUE)$se.fit
[1] 0.005227347
```

Categorical effects/dummy variables

To represent qualitative factors in multiple regression, we use dummy, binary, or indicator variables.

- temporal effects (1 if Holiday season, 0 if not)
- spatial (1 if in Midwest, 0 if not)

If a factor X takes R possible levels, we use $R-1$ dummies

- Allow the intercept to shift by taking on the value 0 or 1
- $\mathbb{1}_{[X=r]} = 1$ if $X = r$, 0 if $X \neq r$.

$$\mathbb{E}[Y|X] = \beta_0 + \beta_1 \mathbb{1}_{[X=2]} + \beta_2 \mathbb{1}_{[X=3]} + \cdots + \beta_{R-1} \mathbb{1}_{[X=R]}$$

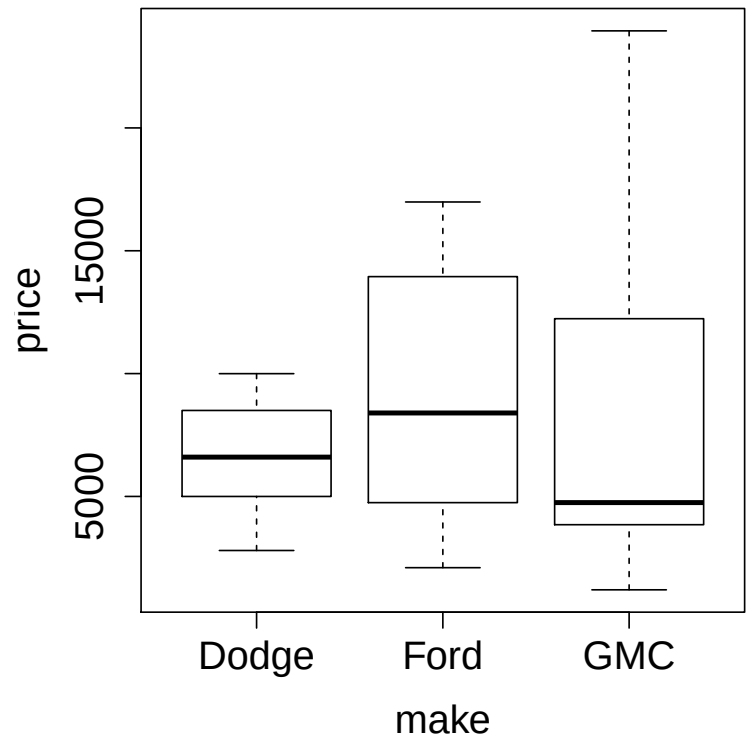
What is $\mathbb{E}[Y|X=1]$?

Example: back to the pickup truck data.

Does `price` vary by `make`?

```
> attach(pickup <- read.csv("pickup.csv"))
> c(mean(price[make=="Dodge"]),
    mean(price[make=="Ford"]),
    mean(price[make=="GMC"]))
[1] 6554.200 8867.917 7996.208
```

- GMC seems lower on average, but lots of overlap.
- Not much of a pattern.



Now fit with linear regression:

$$\mathbb{E}[\texttt{price}|\texttt{make}] = \beta_0 + \beta_1 \mathbb{1}_{[\texttt{make=Ford}]} + \beta_2 \mathbb{1}_{[\texttt{make=GMC}]}$$

Easy in R (if `make` is a factor variable)

```
> summary(trucklm1 <- lm(price ~ make, data=pickup))

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     6554       1787   3.667 0.000671 ***
makeFord        2314       2420   0.956 0.344386
makeGMC         1442       2127   0.678 0.501502
```

The coefficient values correspond to our dummy variables.

What are the p-values?

What if you also want to include mileage?

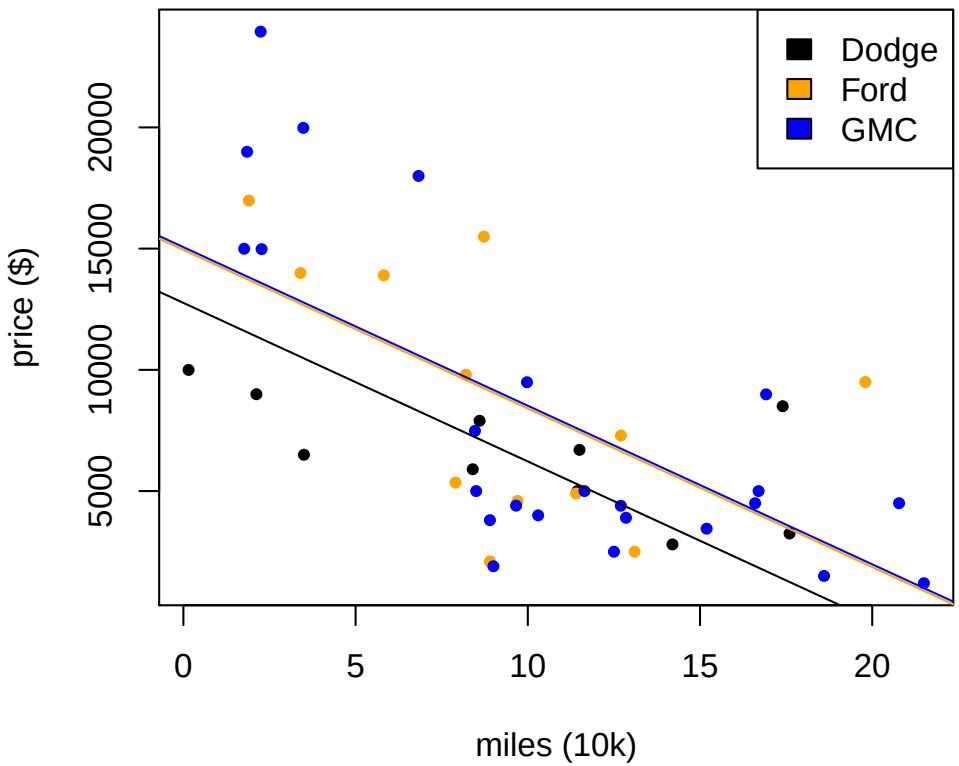
- No problem.

```
> pickup$miles <- pickup$miles/10000
> trucklm2 <- lm(price ~ make + miles, data=pickup)
> summary(trucklm2)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  12761.8     1746.6   7.307 5.31e-09 ***
makeFord      2185.7     1842.9   1.186    0.242
makeGMC       2298.8     1627.0   1.413    0.165
miles         -654.1      115.3  -5.671 1.18e-06 ***
```

All three brands expect to lose \$654 per 10k miles.

Different intercepts, same slope!

```
> plot(miles, price, pch=20, col=make,
     xlab="miles (10k)", ylab="price ($)")
> abline(a=coef(trucklm2)[1],b=coef(trucklm2)[4],col=1)
> abline(a=(coef(trucklm2)[1]+coef(trucklm2)[2]),
    b=coef(trucklm2)[4],col=2)
```



- Dodge trucks affect all slopes!

Variable interaction

So far we have considered the impact of each independent variable in a additive way.

We can extend this notion and include interaction effects through multiplicative terms.

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 (X_{1i} X_{2i}) + \cdots + \varepsilon_i$$

$$\frac{\partial \mathbb{E}[Y|X_1, X_2]}{\partial X_1} = \beta_1 + \beta_3 X_2$$

Interactions with dummy variables

Dummy variables separate out categories

- Different intercept for each category

Interactions with dummies separate out trends

- Different slope for each category

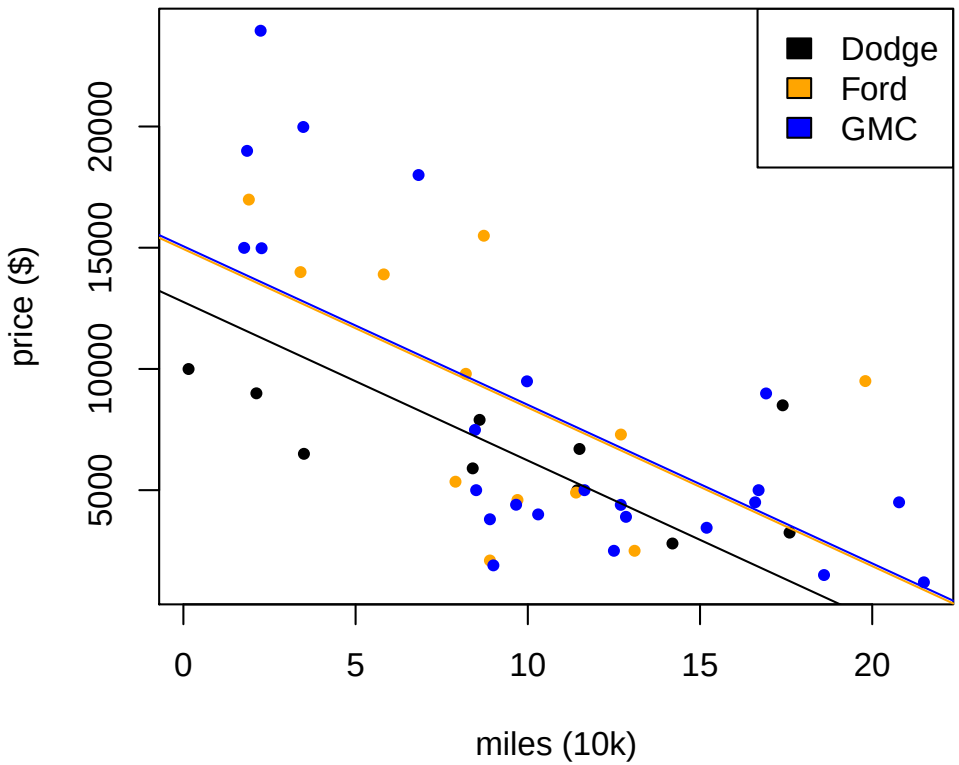
$$Y_i = \beta_0 + \beta_1 \mathbb{1}_{\{X_{1i}=1\}} + \beta_2 X_{2i} + \beta_3 (\mathbb{1}_{\{X_{1i}=1\}} X_{2i}) + \cdots + \varepsilon_i$$

$$\frac{\partial \mathbb{E}[Y|X_1 = 0, X_2]}{\partial X_2} = \beta_2 \qquad \frac{\partial \mathbb{E}[Y|X_1 = 1, X_2]}{\partial X_2} = \beta_2 + \beta_3$$

Same slope, different intercept

- Price difference does not depend on mileage!

```
> trucklm2 <- lm(price ~ make + mile, data=pickup)
```

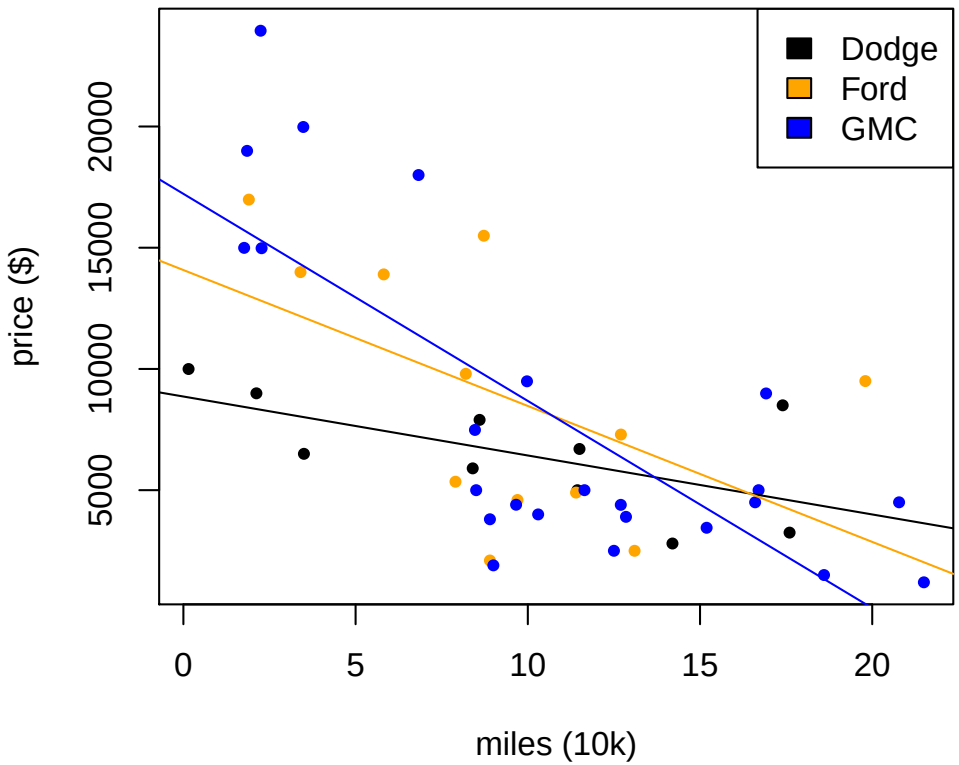


- Dodge trucks affect all slopes!

Now add individual slopes!

- Price difference *varies* with miles!

```
> trucklm3 <- lm(price ~ make*miles, data=pickup)
```



- Dodge doesn't effect Ford, GMC b_0, b_1

What do the numbers show?

```
> summary(trucklm3)

                Estimate Std. Error t value Pr(>|t|)
(Intercept)         8862       2508     3.5    0.001 **
makeFord            5216       3707     1.4    0.167
makeGMC             8360       3080     2.7    0.010 **
miles               -243        225    -1.1    0.287
makeFord:miles      -317        347    -0.9    0.366
makeGMC:miles       -611        268    -2.3    0.028 *

> c(coef(trucklm3)[1], coef(trucklm3)[4]) ##(b_0,b_1) Dodge
(Intercept)       miles
  8862.1987   -243.1789
> c((coef(trucklm3)[1]+coef(trucklm3)[2]), ## b_0 Ford
+   (coef(trucklm3)[4]+coef(trucklm3)[5])) ## b_1 Ford
(Intercept)       miles
 14078.6715   -560.5871
```

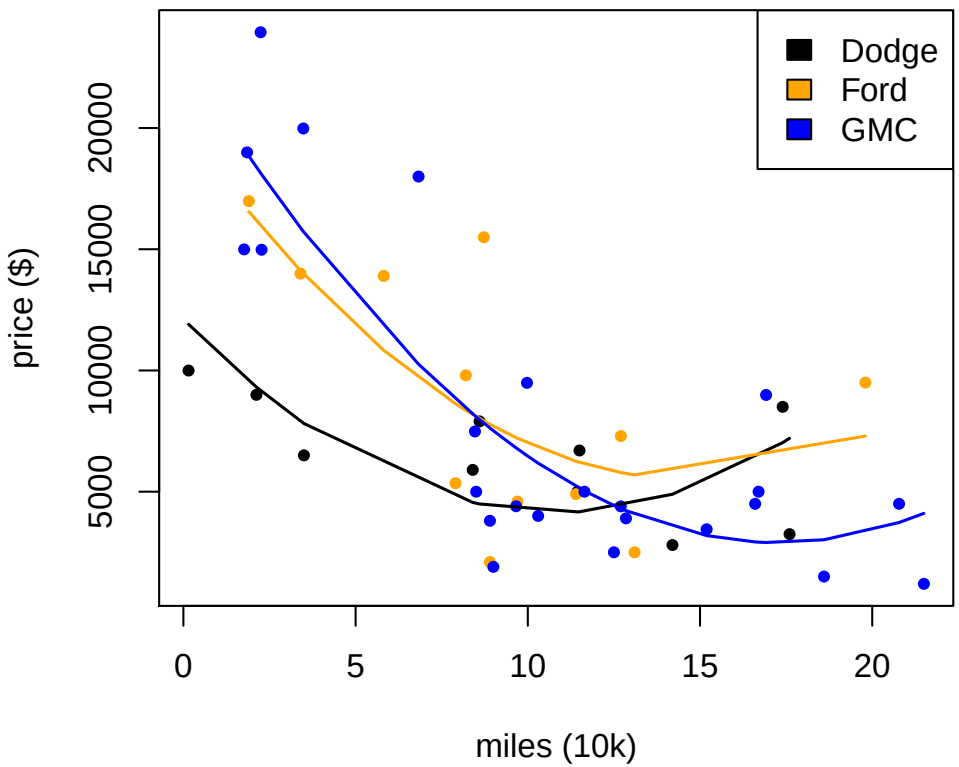
What do the numbers show?

```
> summary(trucklm3)

                Estimate Std. Error t value Pr(>|t|)
(Intercept)         8862       2508     3.5    0.001 **
makeFord            5216       3707     1.4    0.167
makeGMC             8360       3080     2.7    0.010 **
miles               -243        225    -1.1    0.287
makeFord:miles      -317        347    -0.9    0.366
makeGMC:miles       -611        268    -2.3    0.028 *

> price.Ford <- price[make=="Ford"]
> miles.Ford <- miles[make=="Ford"]
> summary(lm(price.Ford ~miles.Ford))

             Estimate Std. Error t value Pr(>|t|)
(Intercept)  14078.7     3094.6   4.549  0.00106 **
miles.Ford    -560.6      299.3  -1.873  0.09054 .
```

Individual slopes, plus X^2!

- Price difference *varies* with miles!

```
> trucklm5 <- lm(price ~ make*miles + I(miles^2), data=pickup)
> see week3-Rcode.R for graphing
```

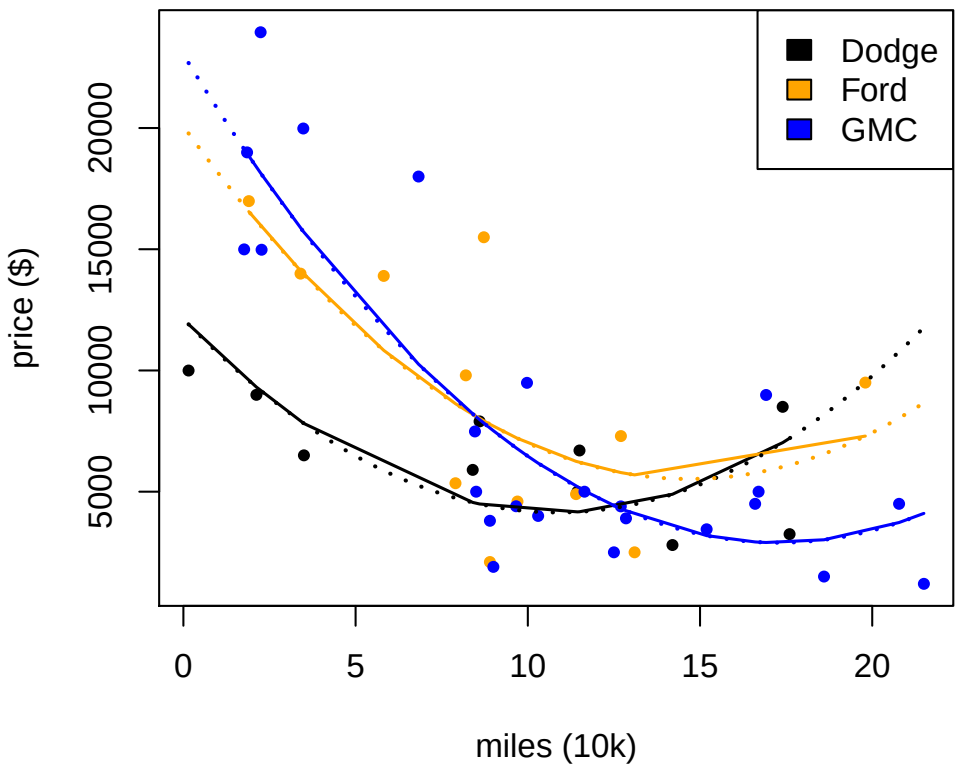


- Common quadratic. Interpretation?
- Extrapolation danger!

Individual slopes, plus X^2!

- Price difference *varies* with miles!

```
> trucklm5 <- lm(price ~ make*miles + I(miles^2), data=pickup)
> see week3-Rcode.R for graphing
```

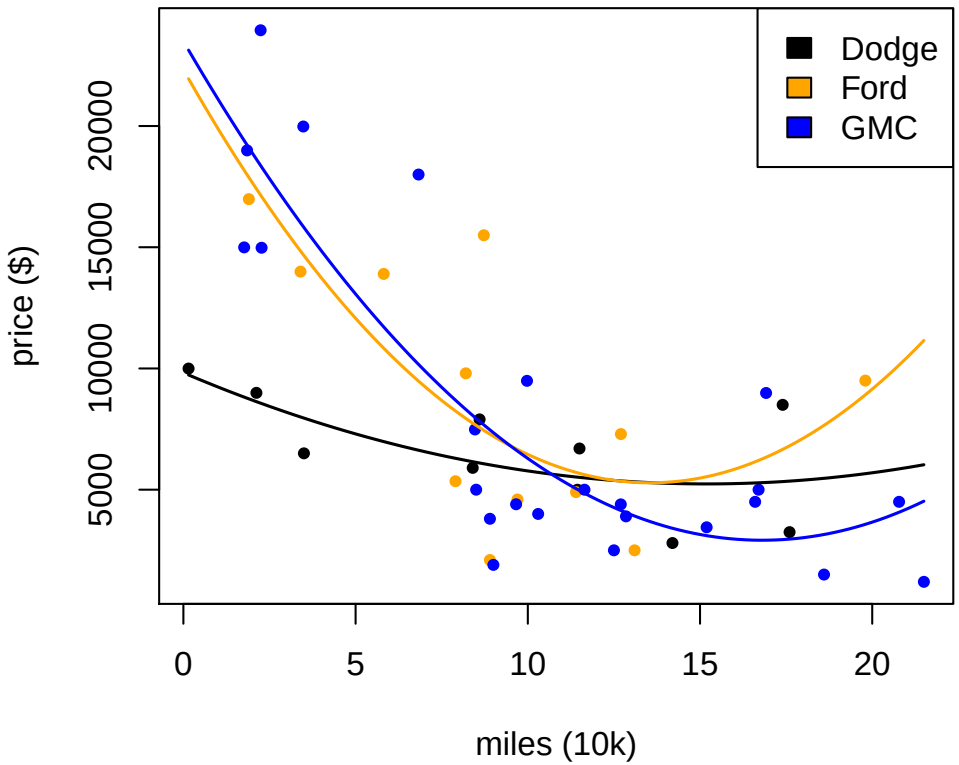


- Common quadratic. Interpretation?
- Extrapolation danger!

Individual slopes and individual X^2!

- Price difference *varies* with miles!

```
> trucklm6 <- lm(price ~ make*(miles+I(miles^2)), data=pickup)
> see week3-Rcode.R for graphing
```



- Different quadratic. Interpretation?
- Extrapolation danger!

Interactions with continuous variables

Example: connection between college & MBA grades. A model to predict Booth GPA from college GPA could be

$$\text{GPA}^{\text{MBA}} = \beta_0 + \beta_1 \text{GPA}^{\text{Bach}} + \varepsilon.$$

```
> grades <- read.csv("grades.csv")
> summary(grades) #output not shown
> attach(grades)
> summary(lm(MBAGPA ~ BachGPA)) ## severly abbrev.
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.58985    0.31206   8.299  1.2e-11 ***
BachGPA      0.26269    0.09244   2.842  0.00607 **
```

- For every 1 point increase in college GPA, your expected GPA at Booth increases by about 0.26 points.

However, this model assumes that the marginal effect of College GPA is the same for any age.

- But I'd guess that how you did in college has less effect on your MBA GPA as you get older (farther from college).

We can account for this intuition with an interaction term:

$$\text{GPA}^{\text{MBA}} = \beta_0 + \beta_1 \text{GPA}^{\text{Bach}} + \beta_2 (\text{Age} \times \text{GPA}^{\text{Bach}}) + \varepsilon$$

Now, the college effect is

$$\frac{\partial \mathbb{E}[\text{GPA}^{\text{MBA}} \mid \text{GPA}^{\text{Bach}}, \text{Age}]}{\partial \text{GPA}^{\text{Bach}}} = \beta_1 + \beta_2 \text{Age}.$$

$\Rightarrow$ Depends on Age!

Fitting interactions in R is easy:

`lm(Y ~ X1*X2)` fits $\mathbb{E}[Y] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$.

Here, we want the interaction but do not want to include the main effect of age (should age matter individually?).

```
> summary(lm(MBAGPA ~ BachGPA*Age - Age))

Coefficients: ## output abbreviated
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.820494   0.296928   9.499 1.23e-13 ***
BachGPA      0.455750   0.103026   4.424 4.07e-05 ***
BachGPA:Age -0.009377   0.002786  -3.366  0.00132 **
```

Without the interaction term

- Marginal effect of College GPA is $b_1 = 0.26$.

With the interaction term:

- Marginal effect is $b_1 + b_2 \text{Age} = 0.46 - 0.0094 \text{Age}$.

Age	Marginal Effect
25	0.22
30	0.17
35	0.13
40	0.08

Summary

Multiple linear regression is just like SLR

. . . with one important tweak.

Interpretation is crucial:

- Polynomials
- Log transformations
- Holding other variables fixed
- Interactions

Coming Up Next

Next Week: More on MLR

- Data issues and different structures

Week 5

- MIDTERM! Hurray!
- Correlation implies causation

Week 6–9: Advertisements for other classes

- discrete outcomes, time series, (baby) machine learning

Glossary and equations

- Model: $Y|X_1, \ldots, X_d \overset{ind}{\sim} \mathcal{N}(\beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d, \sigma^2)$
- Prediction: $\hat{Y}_i = b_0 + b_1 X_{1i} + b_2 X_{2i} + \cdots + b_d X_{di}$
- $\mathbf{b} \sim \mathcal{N}_p(\boldsymbol{\beta}, \mathbf{S_b})$
- Interaction:
  - $Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 (X_{1i} X_{2i}) + \ldots + \varepsilon$
  - $\frac{\partial \mathbb{E}[Y|X_1, X_2]}{\partial X_1} = \beta_1 + \beta_3 X_2$

Elasticity is the slope in a log-log model: $\beta_1 \approx \dfrac{d\%Y}{d\%X}$.

(See handout on course website.)