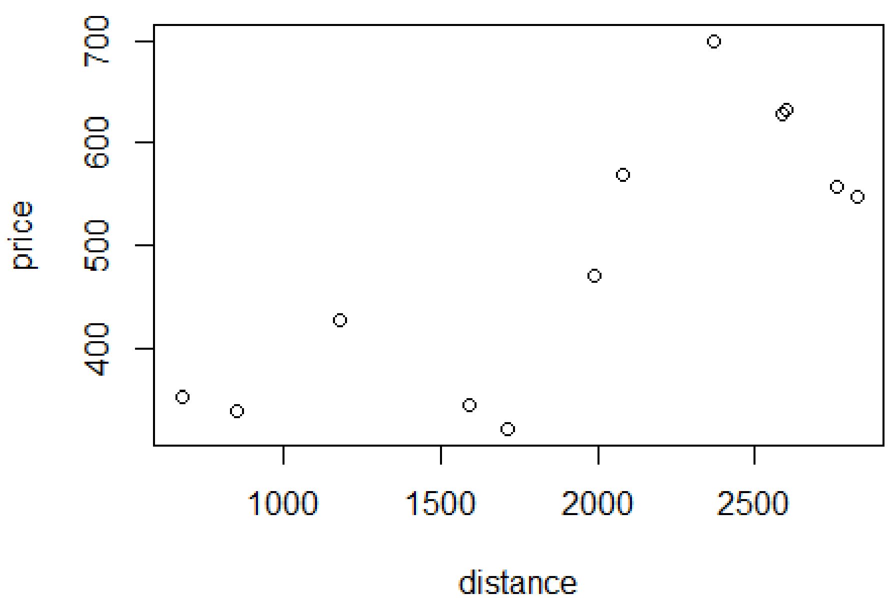
Stat 414 — Day 2
Estimating Linear Models

Last Time

- Multilevel data is when the structure of the data is characterized by “observational units” at different levels, often from clustering or nesting in the data (e.g., students nested in classrooms)
- Multilevel data needs to be analyzed differently from single level data

Example 1: Predicting airfare cont

	price	distance	city
1	632	2604	JacksonvilleFlorida
2	339	850	SaltLakeCityUtah
3	628	2590	CharlotteNorthCarolina
4	353	673	TucsonArizona
5	700	2370	JacksonMississippi
6	471	1990	StLouisMissouri



Scatterplot airfare prices vs. distance

(a) Does it seem reasonable to fit a linear model to these data? How are you deciding?

The overall pattern seems to have a reasonably constant rate of increase in price as distance increases.

Least Squares Estimation

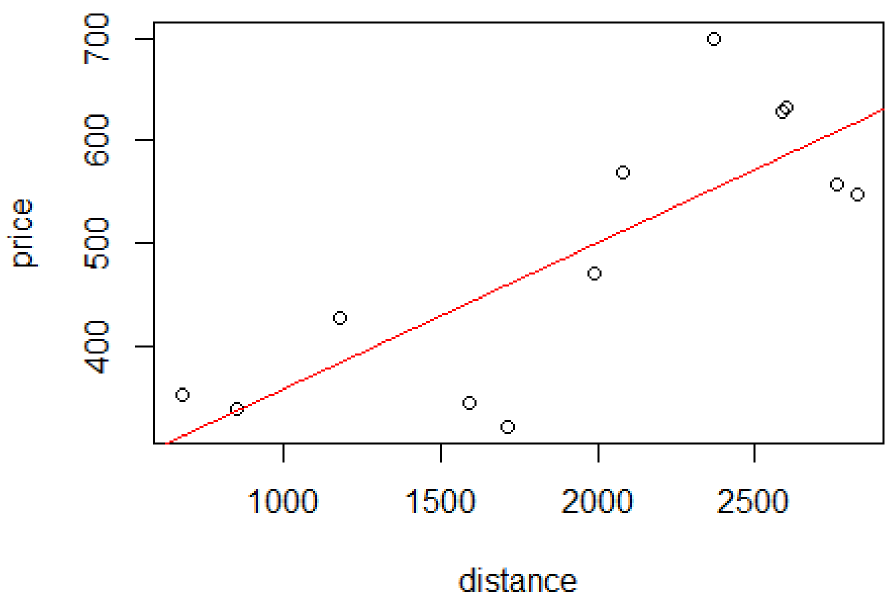
The **least squares regression model** fits the best fitting line by minimizing the sum of the squared residuals.

```
model1 = lm(price ~ distance, data=airfare); model1
```

```
Call:
lm(formula = price ~ distance, data = airfare)

Coefficients:
(Intercept)     distance
   214.994        0.142
```

```
with(airfare, plot(price ~ distance)); abline(model1, col = "red")
```



plot of price vs. distance with OLS line overlaid

A few R tricks

Check out this cool trick

```
coefs <- coef(model1)
```

The intercept of the regression is 214.99 and the slope of the regression is 0.14.

In fact, many times in R we only want to see some of the output, e.g.,

```
summary(model1)$r.squared
[1] 0.64
summary(model1)$sigma
```

```
[1] 83
```

A key metric of "model fit" is the sum of the squared residuals. The residual standard error is the square root of the mean squared error, $\Sigma(y_i - \hat{y}_i)^2/(n-2)$

```
sqrt(sum(residuals(model1)^2/(10)))
[1] 83
```

Interpreting the model

(b) How should we *interpret* the intercept, slope, R^2, and σ values?

R^2 says 63.9% of the variation in prices from SLO is explained by the distance from SLO. /n intercept = 214.99 dollars = predicted price when distance is zero ('set-up cost') /n slope = .14 dolaars per mile = for each one mile increase in distance there is an associated/predict/on average with an .14 dollar increase in price of flight /n 83.46 dollars = typical error between observed price and predicted prices /n

Evaluating the model

One way to *evaluate* the linear model is to fit a more complicated model and see how much better it fits the data.

```
#Add a quadratic term to the model. Use the I() function to square the variable bef
ore running the model
model2 <- lm(price ~ distance + I(distance^2), data = airfare)
model2

Call:
lm(formula = price ~ distance + I(distance^2), data = airfare)

Coefficients:
  (Intercept)       distance  I(distance^2)
    2.82e+02       5.24e-02       2.52e-05
summary(model2)$r.squared
[1] 0.65
sigma(model2)
[1] 87
```

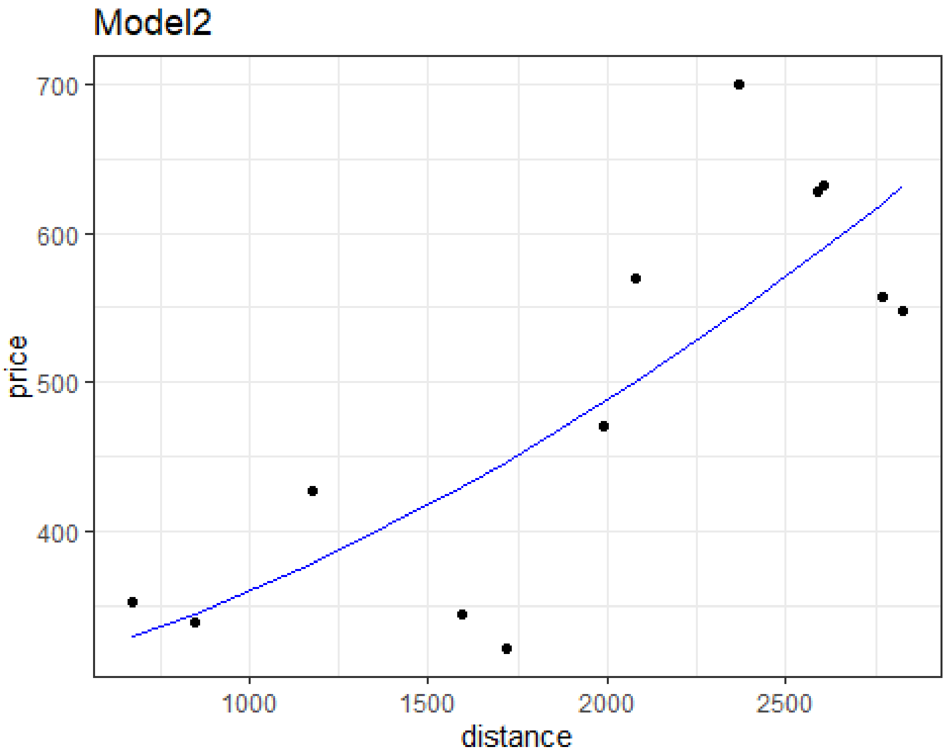
(c) Based on the output, what is the impact of the quadratic term? Is this a better fitting model? How are you deciding?

now have a slight upward curve in the prices as distances increases. Better fitting in term sof R^2 but only slightly.

Always a good habit to examine the model behavior visually as well!

```
#install.packages("tidyverse")
#library(tidyverse)
airfare |>
  ggplot(aes(x = distance, y = price)) +
  geom_point() +
```

```
  geom_line(aes(x = distance, y = model2$fitted.values), color = "blue") +
  labs(title = "Model2") +
 theme_bw()
```



Scatterplot of prices vs. distance with OLS line overlaid

(d) Would you consider this a meaningfully different model? Worth the extra complication in interpreting the model?

only a slight increase in R^2, maybe not worth the extra complication.

Definition
$Adjusted\ R^2$ penalizes the model for requiring estimation of additional parameters.

Compare the R^2 values for the two models.

```
summary(model1)$adj.r.squared
[1] 0.6
summary(model2)$adj.r.squared
[1] 0.57
```

(e) Which model would you recommend and why?
Model 1, higher R^2 adjusted but I would be willing to hear both arguments

Maximum Likelihood Estimation

Maximum likelihood estimation estimates the parameters to maximize the likelihood of seeing your data.

```
#install.packages("nlme")
library(nlme)

Attaching package: 'nlme'
The following object is masked from 'package:dplyr':

   collapse
model1ML <- nlme::gls(price ~ distance, data = airfare, method = "ML")
model1ML
Generalized least squares fit by maximum likelihood
  Model: price ~ distance
  Data: airfare
  Log-likelihood: -69

Coefficients:
(Intercept)    distance
    214.99       0.14

Degrees of freedom: 12 total; 10 residual
Residual standard error: 76
```

(f) How have the estimated values for the intercept and slope changed?
They did not!

Note that a key metric in this output is the value of the log-likelihood when the estimated values are substituted back into the likelihood function. This metric essentially replaces the sum of squared errors in comparing models.

So what about the estimate of σ?

```
logLik(model1ML)
'log Lik.' -69 (df=3)
sigma(model1ML)
[1] 76
```

(g) How has the estimated value for σ changed from the least squares model?
OLS gave us sigma-hat = 86, MLE gave us sigma-hat = 76.2 which is smaller. Will talk more about this soon.

Likelihood estimation also includes a mechanism for "penalizing" your fit statistics based on the number of parameters being estimated (akin to $adjusted\ R^2$)

```
AIC(model1ML)  #-2 x log-likelihood + 2p, p = number of parameters in the model
[1] 144
BIC(model1ML)  #-2 x log-likelihood + p x ln(n)
[1] 146
```

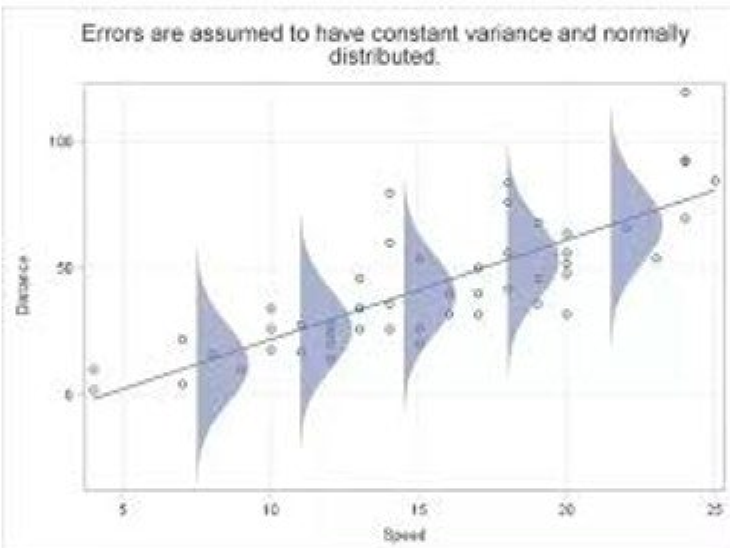
(h) Do we want large values or small values for these?

smaller

Statistical Inference

So far we haven't really made any assumptions other than having a linear relationship between Y and X. The LINE or FINE assumptions you are used to caring so much about are really needed for p-values and confidence intervals.

The **Basic Regression Model** (Least Squares) simple: $E(Y_i) = \beta_0 + \beta_1 x_i + \epsilon_i$ where the ϵ_i are assumed to be normally distribution with mean $E(\epsilon_i)$ = 0 and variance $V(\epsilon_i) = \sigma^2$.



Basic Regression Model