Topics in Multivariate Analysis

APSTA GE-2004

Lecture 2 - Regression and F-Tests

2/1/2022

Outline

Welcome! Today, we'll cover the following:

- Regression and F-Tests
  - Bivariate
  - Sets of coefficients
- Changing units of measurement

Reading:

RAOS 3.4; Ch 8-9; Ch 10.7; Ch 11.1-11.4,11.6; Ch 12.1-12.5; Ch 16.4
User-friendly Bayesian regression modeling by Muth, Oravecz, & Gabry
RAOS Appendix A and B

Regression and F-Tests

Linear regression

Recall our linear model (kid_score = a + b*mom_iq + error), described with the following model components:

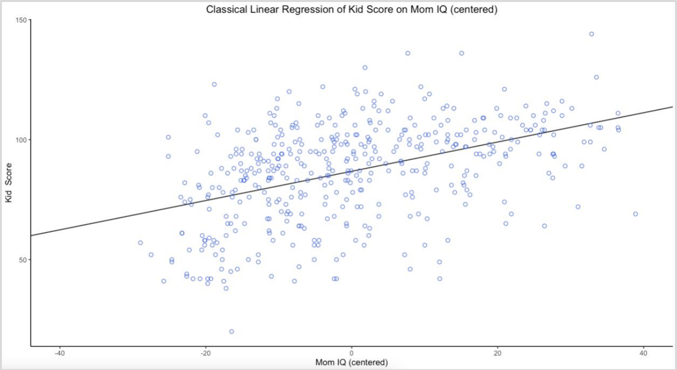
$y_i \sim \text{Normal}(\mu_i, \sigma)$ [likelihood]

$\mu_i = a + b(x_i - \text{mean}(x))$ [linear model]

Linear regression: OLS

Now let's see how to specify this model as a classical regression in R:

```
# Fit a classical linear regression
options(show.signif.stars = FALSE,
        show.coef.Pvalues = FALSE)
fit_ols <- lm(kid_score ~ mom_iq_centered, data=kidiq)
summary( fit_ols )
arm::display( fit_ols )
par(mfrow=c(2,2))
plot(fit_ols)
par(mfrow=c(1,1))
```



Interpreting OLS output

- Residuals
- Coefficients
- Residual standard error
- Adjusted R-squared
- F-statistic

```
Call:
lm(formula = kid_score ~ mom_iq_centered, data = kidiq)

Residuals:
    Min      1Q  Median      3Q     Max 
-56.753 -12.074   2.217  11.710  47.691 

Coefficients:
                Estimate Std. Error t value
(Intercept)     86.79724    0.87680   98.99
mom_iq_centered  0.60997    0.05852   10.42

Residual standard error: 18.27 on 432 degrees of freedom
Multiple R-squared:  0.201,	Adjusted R-squared:  0.1991 
F-statistic: 108.6 on 1 and 432 DF,  p-value: < 2.2e-16
```

Interpreting OLS output: Residuals

Residuals (difference between actual and predicted): $\boxed{r_i = y_i - \hat{y}_i}$

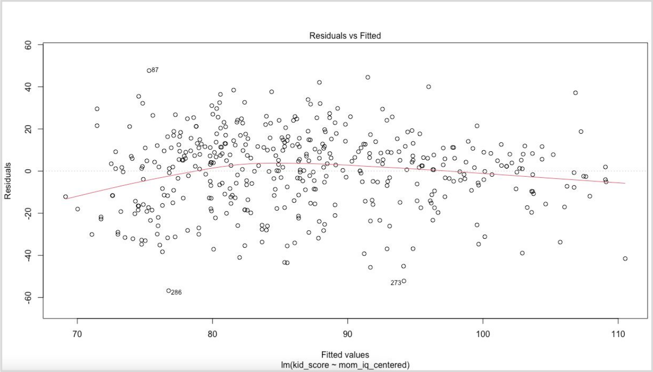
- The average of the residuals is zero by definition, so the median should not be far from zero
- **Top right**: If model isn't misspecified, residuals should look roughly randomly scattered about the horizontal line
- **Bottom right**: If residuals are normally distributed, they should fall along the 45° line

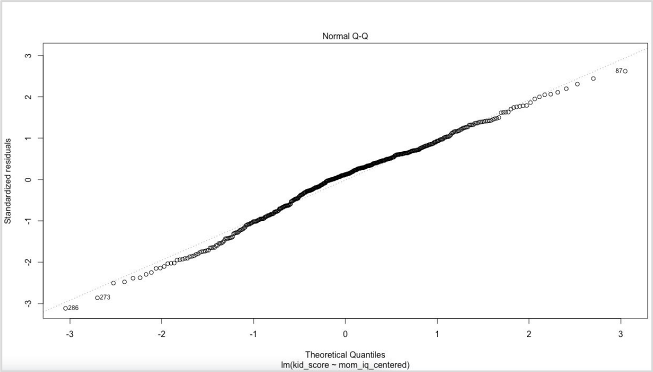
```
Call:
lm(formula = kid_score ~ mom_iq_centered, data = kidiq)

Residuals:
    Min      1Q  Median      3Q     Max 
-56.753 -12.074   2.217  11.710  47.691 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)     86.79724    0.87680   98.99   <2e-16 ***
mom_iq_centered  0.60997    0.05852   10.42   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 18.27 on 432 degrees of freedom
Multiple R-squared:  0.201,	Adjusted R-squared:  0.1991 
F-statistic: 108.6 on 1 and 432 DF,  p-value: < 2.2e-16
```



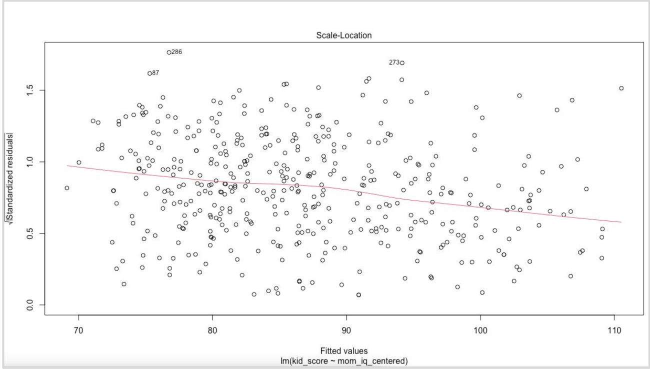


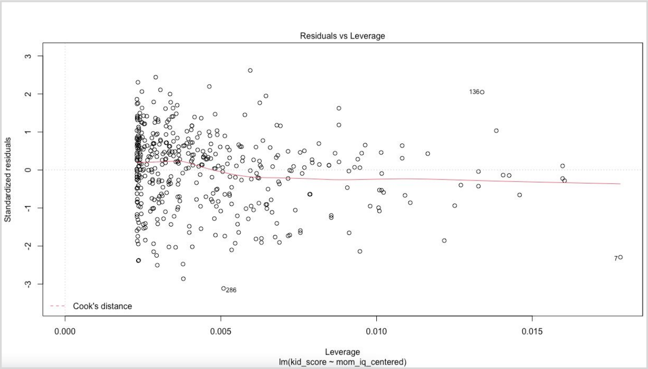
Interpreting OLS output: Residuals

Top right: If residuals are homoscedastic (i.e. constant variance), they should spread equally along a horizontal line over the range of the predictor

Bottom right: the plot provides information about individual observations, especially outliers, high-leverage points, and influential observations:

- **outlier**: an observation that isn't predicted well by the fitted regression model (i.e. has a large positive or negative residual)
- **high-leverage point**: has an unusual combination of predictor values. That is, it's an outlier in the predictor space
- **influential observation**: has a disproportionate impact on the determination of the model parameters (identified using a statistic called Cook's distance, or Cook's D)





Interpreting OLS output: Coefficients

The fitted model is:

kid_score = 87 + 0.6 * mom_iq_centered + error

Interpreting points on fitted line:

Either as predicted test scores for kids at each of several maternal IQ levels, or as average test scores for subpopulations defined by these scores

Intercept:

Since we centered mom IQ, the intercept reflects the predicted test scores of kids whose mothers have average IQ (100)

Slope:

If we compare average kid scores for subpopulations that differ in maternal IQ by 1 point, we expect to see that the group with higher maternal IQ achieves 0.6 points more on average
(if IQs differ by 10 points, scores differ by 6 points on average)

```
Call:
lm(formula = kid_score ~ mom_iq_centered, data = kidiq)

Residuals:
    Min      1Q  Median      3Q     Max 
-56.753 -12.074   2.217  11.710  47.691 

Coefficients:
                Estimate Std. Error t value
(Intercept)     86.79724    0.87680   98.99
mom_iq_centered  0.60997    0.05852   10.42

Residual standard error: 18.27 on 432 degrees of freedom
Multiple R-squared:  0.201,     Adjusted R-squared:  0.1991 
F-statistic: 108.6 on 1 and 432 DF,  p-value: < 2.2e-16
```

Interpreting OLS output: Coefficient Estimates

The fitted model is:

kid_score = 87 + 0.6 * mom_iq_centered + error

Slope:

$b_1 = \Sigma(x_i - \bar{x})(y_i - \bar{y}) / \Sigma(x_i - \bar{x})^2$

= sample covariance between x_i and y_i divided by the sample variance of x_i

$= r * (\sigma_y / \sigma_x)$

= r (sample correlation between x_i and y_i) and σ_x and σ_y are sample std devs

Intercept:

$b_0 = \bar{y} - b_1 * \bar{x}$

```
Call:
lm(formula = kid_score ~ mom_iq_centered, data = kidiq)

Residuals:
    Min      1Q  Median      3Q     Max
-56.753 -12.074   2.217  11.710  47.691

Coefficients:
                Estimate Std. Error t value
(Intercept)     86.79724    0.87680   98.99
mom_iq_centered  0.60997    0.05852   10.42

Residual standard error: 18.27 on 432 degrees of freedom
Multiple R-squared:  0.201,     Adjusted R-squared:  0.1991
F-statistic: 108.6 on 1 and 432 DF,  p-value: < 2.2e-16
```

```
# Coefficients: b0 and b1
# b1
( b1 <- sum( (mom_iq_centered - mean(mom_iq_centered)) * (kidiq$kid_score - mean(kidiq$kid_score)) ) /
  sum( (mom_iq_centered - mean(mom_iq_centered))^2 ) )
# b0
mean(kidiq$kid_score) - (b1 * mean(mom_iq_centered))
```

Interpreting OLS output: Coefficient Standard Errors

The fitted model is:

kid_score = 87 + 0.6 * mom_iq_centered + error

Slope:

$SE(b_1) = s_e / \sqrt{[TSS_x]}$

$SE(b_1) = \text{sqrt} [\Sigma(y_i - \hat{y}_i)^2 / (n - K)] / \text{sqrt} [\Sigma(x_i - x\_bar)^2]$

Intercept:

$SE(b_0) = s_e * \sqrt{[(1/n) + (x\_bar^2 / TSS_x)]}$

$SE(b_0) = \text{sqrt} [\Sigma(y_i - \hat{y}_i)^2 / (n - K)] *$
$\text{sqrt} [(1/n) + (x\_bar^2 / \Sigma(x_i - x\_bar)^2)]$

```
Call:
lm(formula = kid_score ~ mom_iq_centered, data = kidiq)

Residuals:
    Min      1Q  Median      3Q     Max 
-56.753 -12.074   2.217  11.710  47.691 

Coefficients:
                Estimate Std. Error t value
(Intercept)     86.79724    0.87680   98.99
mom_iq_centered  0.60997    0.05852   10.42

Residual standard error: 18.27 on 432 degrees of freedom
Multiple R-squared:  0.201,	Adjusted R-squared:  0.1991 
F-statistic: 108.6 on 1 and 432 DF,  p-value: < 2.2e-16
```

```
# standard error of b0
sqrt( sum( (kidiq$kid_score - fit_ols$fitted.values)^2 ) / (434-2) ) *
  sqrt( (1/434) + ( mean(mom_iq_centered)^2 / sum( (mom_iq_centered - mean(mom_iq_centered))^2 ) ) )

# standard error of b1
sqrt( sum( (kidiq$kid_score - fit_ols$fitted.values)^2 ) / (434-2) ) /
  sqrt( sum( (mom_iq_centered - mean(mom_iq_centered))^2 ) )
```

Interpreting OLS output: Coefficient t-statistics

The fitted model is:

kid_score = 87 + 0.6 * mom_iq_centered + error

t-statistics:

$t = (b - \beta) / SE_b$

Assuming $\beta = 0$, it simplifies to:

$t = b / SE_b$ with (n - K) degrees of freedom

For example:

$\text{t-value}_{b1} = 0.60997 / 0.05852 = 10.42$

(n - K) = 434 - 2 = 432 degrees of freedom

```
Call:
lm(formula = kid_score ~ mom_iq_centered, data = kidiq)

Residuals:
    Min      1Q  Median      3Q     Max 
-56.753 -12.074   2.217  11.710  47.691 

Coefficients:
                Estimate Std. Error t value
(Intercept)     86.79724    0.87680   98.99
mom_iq_centered  0.60997    0.05852   10.42

Residual standard error: 18.27 on 432 degrees of freedom
Multiple R-squared:  0.201,	Adjusted R-squared:  0.1991 
F-statistic: 108.6 on 1 and 432 DF,  p-value: < 2.2e-16
```

Interpreting OLS output: Residual standard error

The fitted model is:

kid_score = 87 + 0.6 * mom_iq_centered + error

```
Call:
lm(formula = kid_score ~ mom_iq_centered, data = kidiq)

Residuals:
    Min      1Q  Median      3Q     Max 
-56.753 -12.074   2.217  11.710  47.691 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)     86.79724    0.87680   98.99   <2e-16 ***
mom_iq_centered  0.60997    0.05852   10.42   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 18.27 on 432 degrees of freedom
Multiple R-squared:  0.201,	Adjusted R-squared:  0.1991 
F-statistic: 108.6 on 1 and 432 DF,  p-value: < 2.2e-16
```

RSS: $\Sigma(y_i - \hat{y}_i)^2$

RSE: sqrt[$\Sigma(y_i - \hat{y}_i)^2$ / (n – K)]

For example:

RSE = sqrt(144137.3 / 432)

= sqrt(333.65)

= 18.27

```
# residual standard error
sqrt( sum( (y - fit_ols$fitted.values)^2 ) / (434 - 2) )
```

Interpreting OLS output: Adjusted R-squared

R-squared (aka coefficient of determination)

R^2 = explained variance / total variance

$= \Sigma(\hat{y}_i – y\_bar)^2 / \Sigma(y_i – y\_bar)^2$

R^2 doesn't account for the complexity of the model (it never decreases with additional predictors)

Adjusted R-squared

$R_a^2 = R^2$ - [(K - 1) / (n - K) (1 - R^2)]

$= [\ \Sigma(\hat{y}_i – y\_bar)^2 / \Sigma(y_i – y\_bar)^2\ ]$ - [(K - 1) / (n - K) (1 - R^2)]

R_a^2 accounts for the complexity of the model relative to the complexity of the data

```
Call:
lm(formula = kid_score ~ mom_iq_centered, data = kidiq)

Residuals:
    Min      1Q  Median      3Q     Max
-56.753 -12.074   2.217  11.710  47.691

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     86.79724    0.87680   98.99   <2e-16 ***
mom_iq_centered  0.60997    0.05852   10.42   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 18.27 on 432 degrees of freedom
Multiple R-squared:  0.201,     Adjusted R-squared:  0.1991
F-statistic: 108.6 on 1 and 432 DF,  p-value: < 2.2e-16
```

```
# R-squared
y_bar <- mean( kidiq$kid_score , na.rm = TRUE )
( r.squared <- sum( (fit_ols$fitted.values - y_bar)^2 ) / sum( (y - y_bar)^2 ) )

# Adjusted R-squared
( r.squared.adj <- r.squared - ( (2-1) / (434-2)*(1-r.squared) ) )
```

Interpreting OLS output: F-statistic

F-statistic

The F-test of overall significance indicates whether the regression model provides a better fit to the data than a model that contains no predictor variables

F = explained sum of squares/(K - 1) / residual sum of squares/(n - K)

= ESS/(K - 1) / RSS/(n - K)

$= [\Sigma(\hat{y}_i - y\_bar)^2 / (K - 1)] / [\Sigma(y_i - \hat{y}_i)^2 / (n - K)]$

In bivariate regression the F- and two-sided t-tests are redundant. Their relationship is:

$F = t^2$

In multiple regression, F-statistics can test more complex hypotheses regarding *sets* of coefficients

```
# F-statistic
( sum( (fit_ols$fitted.values - y_bar)^2 ) / (2-1) ) /
( sum( (y - fit_ols$fitted.values)^2 ) / (434-2) )
```

```
Call:
lm(formula = kid_score ~ mom_iq_centered, data = kidiq)

Residuals:
    Min      1Q  Median      3Q     Max
-56.753 -12.074   2.217  11.710  47.691

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     86.79724    0.87680   98.99   <2e-16 ***
mom_iq_centered  0.60997    0.05852   10.42   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 18.27 on 432 degrees of freedom
Multiple R-squared:  0.201,	Adjusted R-squared:  0.1991
F-statistic: 108.6 on 1 and 432 DF,  p-value: < 2.2e-16
```

```
> # Critical values for this F distribution
> ( crit10  <- qf( 0.90  , df1 = 1 , df2 = 432 ) )
[1] 2.7
> ( crit05  <- qf( 0.95  , df1 = 1 , df2 = 432 ) )
[1] 3.9
> ( crit025 <- qf( 0.975 , df1 = 1 , df2 = 432 ) )
[1] 5.1
> ( crit01  <- qf( 0.99  , df1 = 1 , df2 = 432 ) )
[1] 6.7
> ( crit001 <- qf( 0.999 , df1 = 1 , df2 = 432 ) )
[1] 11
```

F-Test
(Bivariate regression)

Bivariate regression

We can describe our linear model (water_use_1981 = a + b*income + error) with the following model components:

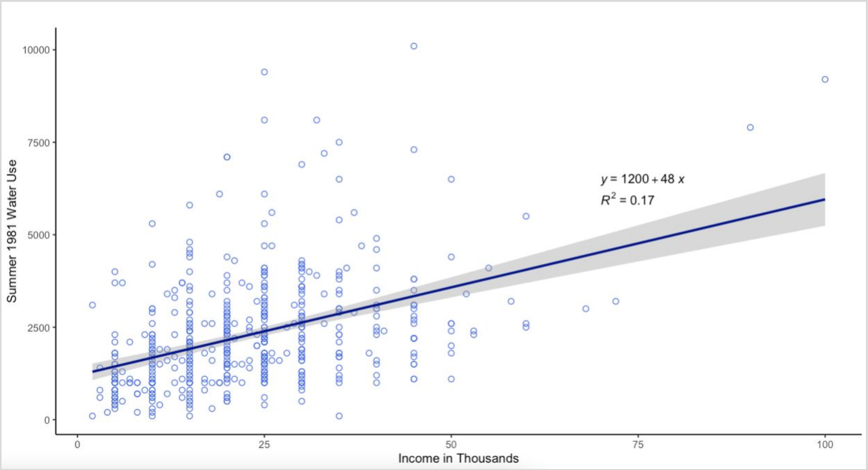
$y_i \sim \text{Normal}(\mu_i, \sigma)$ [likelihood]

$\mu_i = a + b*x_i$ [linear model]

Bivariate regression: OLS

Now let's see how to specify this model as a classical regression in R:

```
# Fit a classical linear regression
options(show.signif.stars = FALSE,
        show.coef.Pvalues = FALSE)
fit_ols <- lm(water81 ~ income, data=concord1)
summary( fit_ols )
arm::display( fit_ols )
par(mfrow=c(2,2))
plot(fit_ols)
par(mfrow=c(1,1))
```



Interpreting Coefficients

The fitted model is:

water_use_1981 = 1201 + 48 * income + error

Interpreting points on fitted line:

Either as predicted water use for households at each of several income levels, or as average water use for subpopulations defined by these income levels

Intercept:

The intercept reflects the predicted water use of households whose income is 0. This is not a useful prediction, since no households have an income of 0. We should center the predictor to make the interpretation of the intercept more meaningful

Slope:

If we compare average water use for households that differ in income by $1,000, we expect to see that the group with higher income uses 48 cubic feet more water on average

```
> summary( fit_ols )

Call:
lm(formula = water81 ~ income, data = concord1)

Residuals:
    Min      1Q  Median      3Q     Max
-2765.3  -889.8  -239.8   536.8  7010.2

Coefficients:
            Estimate Std. Error t value
(Intercept) 1201.124    123.325    9.74
income        47.549      4.652   10.22

Residual standard error: 1352 on 494 degrees of freedom
Multiple R-squared:  0.1745,    Adjusted R-squared:  0.1729
F-statistic: 104.5 on 1 and 494 DF,  p-value: < 2.2e-16

> arm::display( fit_ols )
lm(formula = water81 ~ income, data = concord1)
            coef.est coef.se
(Intercept) 1201.12  123.32
income        47.55    4.65
---
n = 496, k = 2
residual sd = 1351.58, R-Squared = 0.17
```

Interpreting F-statistic

F-statistic

The F-test of overall significance indicates whether the regression model provides a better fit to the data than a model that contains no predictor variables

F = explained sum of squares/(K - 1) / residual sum of squares/(n - K)
= ESS/(K - 1) / RSS/(n - K)
= $[\Sigma(\hat{y}_i - y\_bar)^2 / (K - 1)] / [\Sigma(y_i - \hat{y}_i)^2 / (n - K)]$

In bivariate regression the F- and two-sided t-tests are redundant. Their relationship is:

$F = t^2$

In multiple regression, F-statistics can test more complex hypotheses regarding *sets* of coefficients

```
> # F-statistic
> y <- concord1$water81
> y_bar <- mean( concord1$water81 , na.rm = TRUE )
>
> ( F_statistic <- ( sum( (fit_ols$fitted.values - y_bar)^2 ) / (2-1) ) /
+   ( sum( (y - fit_ols$fitted.values)^2 ) / (496-2) ) )
[1] 104.4586
>
> # In bivariate regression, F = t^2
> res <- summary( fit_ols )
> income_tval <- res$coefficients[2, "t value"]
> income_tval^2
[1] 104.4586
```

```
Call:
lm(formula = water81 ~ income, data = concord1)

Residuals:
    Min      1Q  Median      3Q     Max 
-2765.3  -889.8  -239.8   536.8  7010.2 

Coefficients:
            Estimate Std. Error t value
(Intercept) 1201.124    123.325    9.74
income        47.549      4.652   10.22

Residual standard error: 1352 on 494 degrees of freedom
Multiple R-squared:  0.1745,    Adjusted R-squared:  0.1729 
F-statistic: 104.5 on 1 and 494 DF,  p-value: < 2.2e-16
```

F-Test
(Multiple regression)

F-Test

A regression with K-1 predictor variables requires K parameter estimates: one on each predictor, plus a Y-intercept

The t-statistics test hypotheses regarding individual parameters

The F-statistics can test hypotheses regarding *sets* of parameters. They do this by comparing *nested* models: two models, one a subset of the other

F-Test

We test whether a complex model, with K parameters, significantly improves upon a simpler model with H fewer parameters (0 < H < K):

$$\mathbf{F^{H}_{n\text{-}K} = [\ (RSS\{K - H\} - RSS\{K\})\ /\ H\ ]\ /\ [\ (RSS\{K\})\ /\ (n - K)\ ]}$$

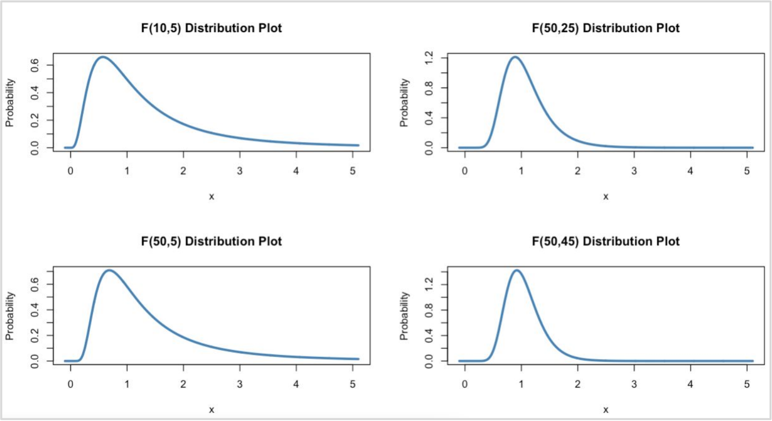
where RSS{K} denotes the residual sum of squares for the complex (K parameters) model, and RSS{K – H} is the residual sum of squares for a model with K – H parameters

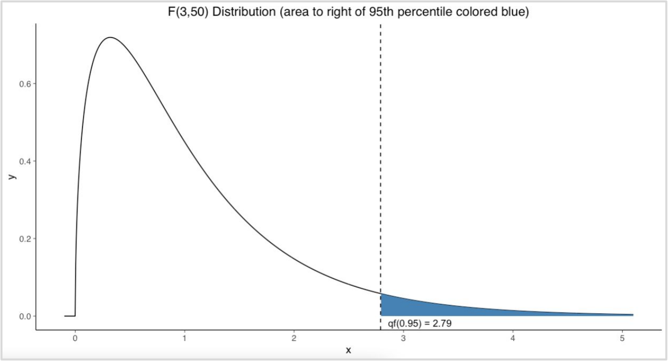
We compare these F-statistics to a theoretical F-distribution with $df_1 = H$ and $df_2 = n - K$ degrees of freedom

F-distributions

The shape of the F distribution depends on both df_1 and df_2. Despite slight differences, the F distributions are all centered on 1

This steadiness in the F distribution makes it easy to informally interpret an F-statistic by eye since a value near 1 is a plausible outcome from the null hypothesis





F-Test
(Multiple regression)

Example 1

Multiple regression

Predictors:

- x_1 - household income, in thousands of dollars
- x_2 - preshortage water use, in cubic feet
- x_3 - education of household head, in years
- x_4 - retirement, coded 1 if household head is retired and 0 otherwise
- x_5 - number of people living in household at time of water shortage (summer 1981)
- x_6 - change in the number of people, summer 1981 minus summer 1980

The variables were chosen from a set of background characteristics thought to predict household water use

Unrestricted (larger) model

We can describe our linear model (water_use_1981 = a + b_1*income + b_2*water80 + b_3*educat + b_4*retire + b_5*peop81 + b_6*cpeop + error) with the following model components:

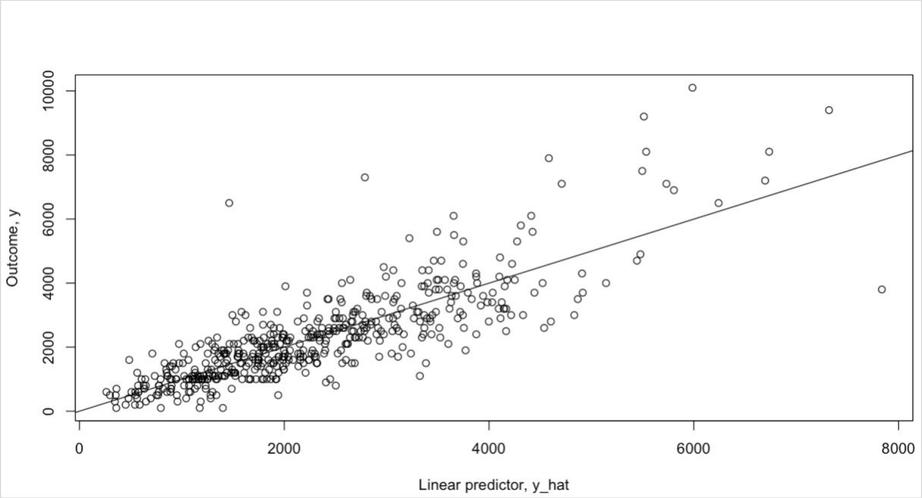
$y_i \sim \text{Normal}(\mu_i, \sigma)$ [likelihood]

$\mu_i = a + b_1 * x_{1i} + b_2 * x_{2i} + b_3 * x_{3i} + b_4 * x_{4i} + b_5 * x_{5i} + b_6 * x_{6i}$ [linear model]

Unrestricted model: OLS

Now let's see how to specify this model as a classical regression in R:

```
# Fit a classical linear regression
options(show.signif.stars = FALSE,
        show.coef.Pvalues = FALSE)
fit_ols <- lm(water81 ~ income + water80 + educat + retire + peop81 + cpeop,
              data=concord1)
summary( fit_ols )
arm::display( fit_ols )
par(mfrow=c(2,2))
plot(fit_ols)
par(mfrow=c(1,1))
```



Interpreting Output

The fitted model is:

water_use_1981 = 242 + 21*income
+ 0.49*water80
– 42*educat
+ 189*retire
+ 248*peop81
+ 96*cpeop
+ error

These six variables together explain about 67% of the variation in postshortage water use (Adjusted R-squared: 0.67)

Now, to understand how to use the F-Test to evaluate the usefulness of sets of coefficients, let's estimate a smaller, *restricted* model that only includes a subset of these variables

```
> summary( fit_ols )

Call:
lm(formula = water81 ~ income + water80 + educat + retire + peop81 +
    cpeop, data = concord1)

Residuals:
    Min      1Q  Median      3Q     Max
-4037.0  -447.6   -69.5   365.4  5038.0

Coefficients:
             Estimate Std. Error t value
(Intercept) 242.22043  206.86382   1.171
income       20.96699    3.46372   6.053
water80       0.49194    0.02635  18.671
educat      -41.86552   13.22031  -3.167
retire      189.18433   95.02142   1.991
peop81      248.19702   28.72480   8.641
cpeop        96.45360   80.51903   1.198

Residual standard error: 849.3 on 489 degrees of freedom
Multiple R-squared:  0.6773,	Adjusted R-squared:  0.6734
F-statistic: 171.1 on 6 and 489 DF,  p-value: < 2.2e-16

> arm::display( fit_ols )
lm(formula = water81 ~ income + water80 + educat + retire + peop81 +
    cpeop, data = concord1)
            coef.est coef.se
(Intercept) 242.22   206.86
income       20.97     3.46
water80       0.49     0.03
educat      -41.87    13.22
retire      189.18    95.02
peop81      248.20    28.72
cpeop        96.45    80.52
---
n = 496, k = 7
residual sd = 849.35, R-Squared = 0.68
```

Restricted (smaller) model

Both education and income reflect socioeconomic status. Suppose we wish to test the hypothesis that socioeconomic status has no linear relationship with water use; that is, we want to test the null hypothesis H_O: $\beta_1 = \beta_3 = 0$

We can describe our linear model (water_use_1981 = a + b_1*water80 + b_2*retire + b_3*peop81 + b_4*cpeop + error) with the following model components:

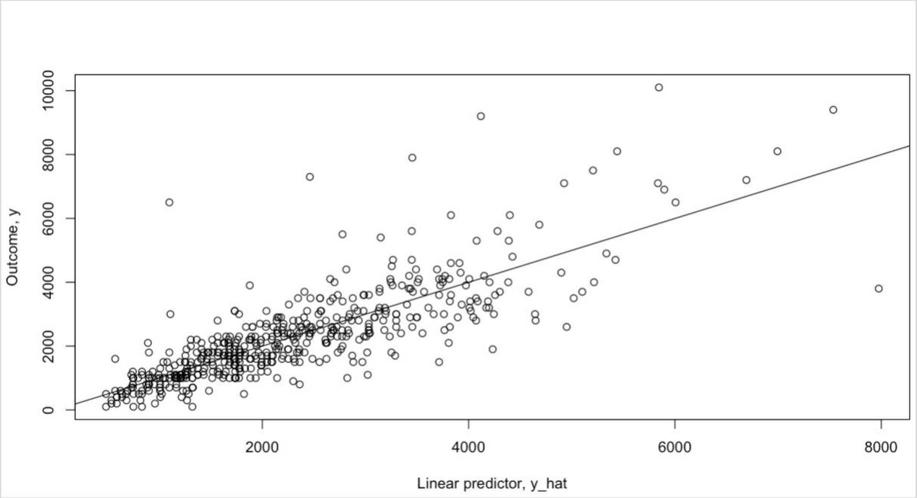
$y_i \sim \text{Normal}(\mu_i, \sigma)$ [likelihood]

$\mu_i = a + b_1 * x_{1i} + b_2 * x_{2i} + b_3 * x_{3i} + b_4 * x_{4i}$ [linear model]

Restricted model: OLS

Now let's see how to specify this model as a classical regression in R:

```
# Fit a classical linear regression
options(show.signif.stars = FALSE,
        show.coef.Pvalues = FALSE)
fit_ols <- lm(water81 ~ water80 + retire + peop81 + cpeop,
              data=concord1)
summary( fit_ols )
arm::display( fit_ols )
par(mfrow=c(2,2))
plot(fit_ols)
par(mfrow=c(1,1))
```



Interpreting Output

The fitted model is:

water_use_1981 = 49 + 0.52*water80
+ 67*retire
+ 265*peop81
+ 134*cpeop
+ error

These four variables together explain about 65% of the variation in postshortage water use (Adjusted R-squared: 0.65)

Now, let's use the F-Test to test whether the unrestricted (larger) model, with 7 parameters, significantly improves upon the restricted (smaller) model with 5 parameters (2 fewer parameters)

```
> summary( fit_ols )

Call:
lm(formula = water81 ~ water80 + retire + peop81 + cpeop, data = concord1)

Residuals:
    Min      1Q  Median      3Q     Max 
-4175.8  -459.5   -78.2   355.7  5401.0 

Coefficients:
             Estimate Std. Error t value
(Intercept)  48.64897  107.05488   0.454
water80       0.51974    0.02677  19.412
retire       67.27992   94.28846   0.714
peop81      265.28936   29.63234   8.953
cpeop       134.46255   83.19590   1.616

Residual standard error: 880.3 on 491 degrees of freedom
Multiple R-squared:  0.6519,	Adjusted R-squared:  0.6491 
F-statistic: 229.9 on 4 and 491 DF,  p-value: < 2.2e-16

> arm::display( fit_ols )
lm(formula = water81 ~ water80 + retire + peop81 + cpeop, data = concord1)
            coef.est coef.se
(Intercept)  48.65   107.05 
water80       0.52     0.03 
retire       67.28    94.29 
peop81      265.29    29.63 
cpeop       134.46    83.20 
---
n = 496, k = 5
residual sd = 880.34, R-Squared = 0.65
```

F-Test

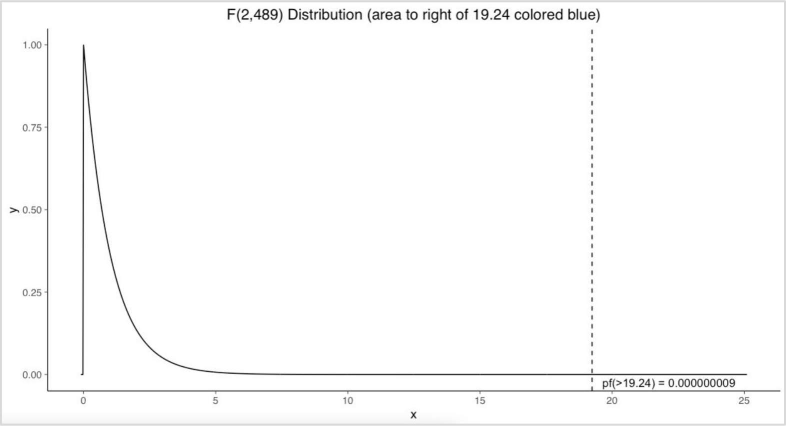
We test whether the unrestricted model, with 7 parameters, significantly improves upon the restricted model with only 5 parameters:

$$\mathbf{F^{H}_{n\text{-}K} = [\ (RSS\{K - H\} - RSS\{K\})\ /\ H\ ]\ /\ [\ (RSS\{K\})\ /\ (n - K)\ ]}$$

$$F^{2}_{489} = [\ (RSS\{5\} - RSS\{7\})\ /\ 2\ ]\ /\ [(RSS\{7\})\ /\ (496 - 7)]$$

$$= [\ (380{,}520{,}363 - 352{,}761{,}188)\ /\ 2\ ]\ /\ [\ 352{,}761{,}188\ /\ 489\ ]$$

$$= \mathbf{19.24}$$



We compare this F-statistic to a theoretical F-distribution with $df_1 = 2$ and $df_2 = 489$ degrees of freedom. This F-statistic leads to a p-value below 0.001. We may reject H_O: $\beta_1 = \beta_3 = 0$

```
> fit_unrestricted <- lm(water81 ~ income + water80 + educat + retire + peop81 + cpeop, data=concord1)
> fit_restricted <- lm(water81 ~ water80 + retire + peop81 + cpeop, data=concord1)
>
> RSS_5 <- sum( (concord1$water81 - fit_restricted$fitted.values)^2 )
> RSS_7 <- sum( (concord1$water81 - fit_unrestricted$fitted.values)^2 )
> H <- 2
> n_minus_K <- 489
>
> ( F_statistic <- ( (RSS_5 - RSS_7) / H ) / ( RSS_7 / n_minus_K ) )
[1] 19.23998
```

F-statistic for a model

As a reminder, the F-statistic reported in the model summary reflects a test of the null hypothesis that *all* of the predictors in a model equal zero. This tests the full model against a model with no predictors and with Y estimated by Y_bar.

For such tests, H = K – 1, and the equation for the F-statistic reduces to:

$$\mathbf{F^{K-1}_{n-K} = [\ (RSS\{1\} - RSS\{K\}) / (K - 1)\ ] / [\ (RSS\{K\}) / (n - K)\ ]}$$

$$= [\ (TSS_Y - RSS) / (K - 1)\ ] / [\ RSS / (n - K)\ ]$$

$$= [\ ESS / (K - 1)\ ] / [\ RSS / (n - K)\ ]$$

```
> # F-statistic for restricted model
> fit_restricted <- lm(water81 ~ water80 + retire + peop81 + cpeop, data=concord1)
>
> ESS <- sum( (fit_restricted$fitted.values - y_bar)^2 )
> RSS <- sum( (concord1$water81 - fit_restricted$fitted.values)^2 )
> K_minus_1 <- (5-1)
> n_minus_K <- (496-5)
>
> ( F_statistic <- ( ESS / K_minus_1 ) / ( RSS / n_minus_K ) )
[1] 229.9119
```

```
> summary( fit_ols )

Call:
lm(formula = water81 ~ water80 + retire + peop81 + cpeop, data = concord1)

Residuals:
    Min      1Q  Median      3Q     Max
-4175.8  -459.5   -78.2   355.7  5401.0

Coefficients:
             Estimate Std. Error t value
(Intercept)  48.64897  107.05488   0.454
water80       0.51974    0.02677  19.412
retire       67.27992   94.28846   0.714
peop81      265.28936   29.63234   8.953
cpeop       134.46255   83.19590   1.616

Residual standard error: 880.3 on 491 degrees of freedom
Multiple R-squared:  0.6519,    Adjusted R-squared:  0.6491
F-statistic: 229.9 on 4 and 491 DF,  p-value: < 2.2e-16

> arm::display( fit_ols )
lm(formula = water81 ~ water80 + retire + peop81 + cpeop, data = concord1)
            coef.est coef.se
(Intercept)  48.65   107.05
water80       0.52     0.03
retire       67.28    94.29
peop81      265.29    29.63
cpeop       134.46    83.20
---
n = 496, k = 5
residual sd = 880.34, R-Squared = 0.65
```

F-Test
(Multiple regression)

Example 2

Multiple regression

Predictors:

- x_1 - years in the league
- x_2 - average games played per year
- x_3 - career batting average
- x_4 - home runs per year
- x_5 - runs batted in per year

The variables were chosen from a set of characteristics thought to predict salary

Unrestricted (larger) model

We can describe our linear model ($\log(salary) = a + b_1{*}years + b_2{*}gamesyr + b_3{*}bavg + b_4{*}hrunsyr + b_5{*}rbisyr + error$) with the following model components:

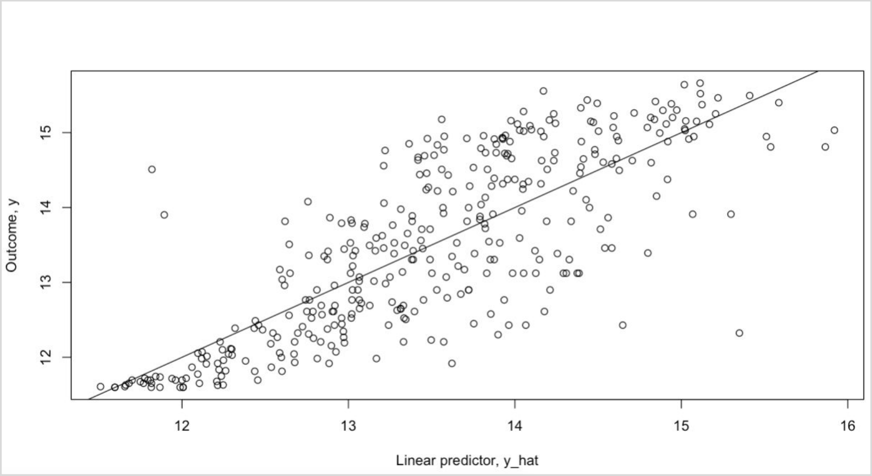
$y_i \sim Normal(\mu_i, \sigma)$ [likelihood]

$\log(\mu_i) = a + b_1{*}x_{1i} + b_2{*}x_{2i} + b_3{*}x_{3i} + b_4{*}x_{4i} + b_5{*}x_{5i}$ [linear model]

Unrestricted model: OLS

Now let's see how to specify this model as a classical regression in R:

```
# Fit a classical linear regression
options(show.signif.stars = FALSE,
        show.coef.Pvalues = FALSE)
mlb1 <- wooldridge::mlb1
fit_unrestricted <- lm(log(salary) ~ years + gamesyr + bavg + hrunsyr + rbisyr,
                       data=mlb1)
options("scipen"=100, "digits"=4)
summary( fit_unrestricted )
options("scipen"=0, "digits"=7)
```



Restricted (smaller) model

Career batting average, home runs per year, and runs batted in per year reflect performance. Suppose we wish to test the hypothesis that performance has no linear relationship with salary; that is, we want to test the null hypothesis H_O: $\beta_3 = \beta_4 = \beta_5 = 0$

We can describe our linear model (log(salary) = a + b_1*years + b_2*gamesyr + error) with the following model components:

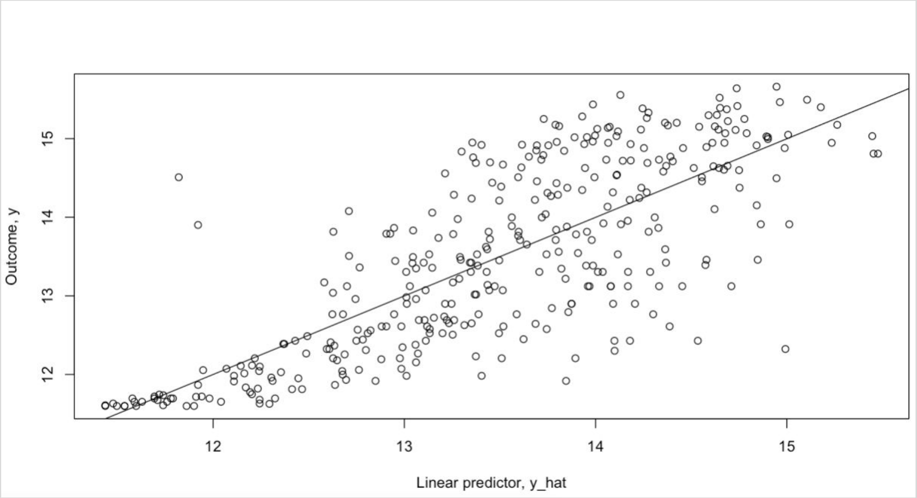
$y_i \sim \text{Normal}(\mu_i, \sigma)$ [likelihood]

$\log(\mu_i) = a + b_1 * x_{1i} + b_2 * x_{2i}$ [linear model]

Restricted model: OLS

Now let's see how to specify this model as a classical regression in R:

```
# Fit a classical linear regression
options(show.signif.stars = FALSE,
        show.coef.Pvalues = FALSE)
mlb1 <- wooldridge::mlb1
fit_restricted <- lm(log(salary) ~ years + gamesyr, data=mlb1)
options("scipen"=100, "digits"=4)
summary( fit_restricted )
options("scipen"=0, "digits"=7)
```



F-Test

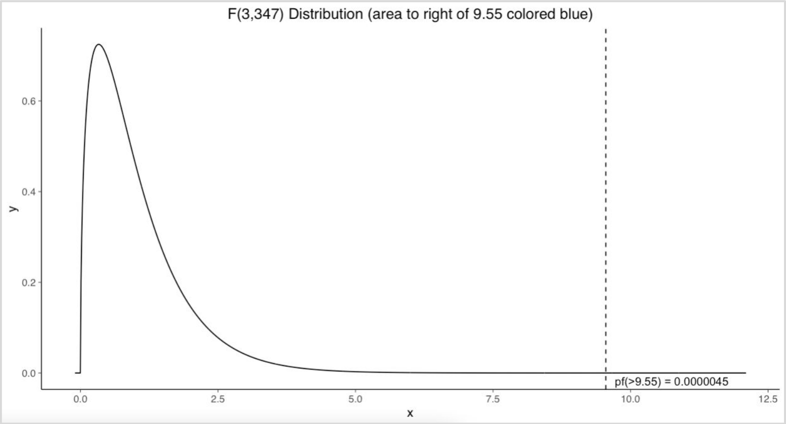
We test whether the unrestricted model, with 6 parameters, significantly improves upon the restricted model with only 3 parameters:

$$\mathbf{F^{H}_{n\text{-}K} = [\ (RSS\{K - H\} - RSS\{K\})\ /\ H\ ]\ /\ [\ (RSS\{K\})\ /\ (n - K)\ ]}$$

$$F^{3}_{347} = [\ (RSS\{3\} - RSS\{6\})\ /\ 3\ ]\ /\ [(RSS\{3\})\ /\ (353 - 6)]$$

$$= [\ (198.3 - 183.1)\ /\ 3\ ]\ /\ [\ 198.3\ /\ 347\ ]$$

$$= \mathbf{9.55}$$



We compare this F-statistic to a theoretical F-distribution with $df_1 = 3$ and $df_2 = 347$ degrees of freedom. This F-statistic leads to a p-value below 0.001. We may reject H_O: $\beta_3 = \beta_4 = \beta_5 = 0$

```
> # Calculate F-statistic for nested models
> fit_unrestricted <- lm(log(salary) ~ years + gamesyr + bavg + hrunsyr + rbisyr, data=mlb1)
> fit_restricted   <- lm(log(salary) ~ years + gamesyr, data=mlb1)
>
> RSS_3 <- sum( (log(mlb1$salary) - fit_restricted$fitted.values)^2 )
> RSS_6 <- sum( (log(mlb1$salary) - fit_unrestricted$fitted.values)^2 )
> H <- 3
> n_minus_K <- (353-6)
>
> ( F_statistic <- ( (RSS_3 - RSS_6) / H ) / ( RSS_6 / n_minus_K ) )
[1] 9.550254
```

Changing Units of Measurement

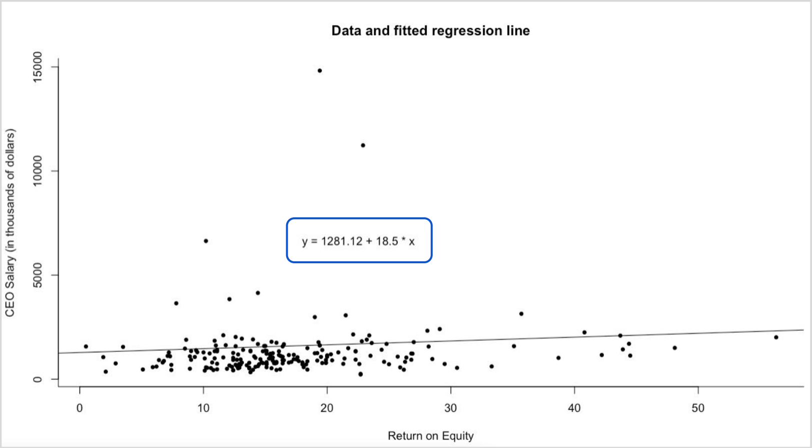
Changing units of measurement: Outcome

Here is a regression of CEO salary (in thousands of dollars) on return on equity (in percent).

If we multiply the outcome variable by a constant *c*, then what happens to the intercept and slope estimates?

For example:
If we change CEO salary (*in thousands of dollars*) to
CEO salary (*in dollars*) by multiplying the variable by 1,000,
then what happens to the intercept and slope estimates?

(assuming nothing about the predictor variable has changed)



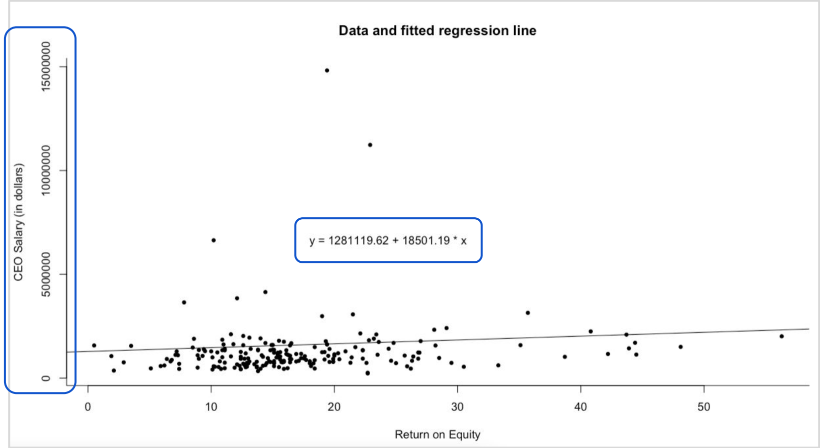

```
> # CEO Salary (in thousands of dollars)
> fit_ceosal1_ols <- lm(salary ~ roe_centered, data = ceosal1)
> tidy(fit_ceosal1_ols)
# A tibble: 2 x 5
  term         estimate std.error statistic  p.value
  <chr>           <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    1281.       94.5     13.6  2.26e-30
2 roe_centered     18.5      11.1      1.66 9.78e- 2
```

Changing units of measurement: Outcome

If the outcome variable is multiplied by a constant *c* – which means each value in the sample is multiplied by *c* – **then the intercept and slope estimates are also multiplied by *c***

For example:
Original outcome: CEO Salary (in thousands of dollars)
Original intercept: 1281
Original coefficient: 18.5

Transformed outcome: CEO Salary *** 1000**
Transformed intercept: 1281 *** 1000** = 1,281,120
Transformed coefficient: 18.50 *** 1000** = 18501




```
> # CEO Salary (in dollars)
> fit_ceosal1_ols_dollars <- lm(salary_dollars ~ roe_centered, data = ceosal1)
> tidy(fit_ceosal1_ols_dollars)
# A tibble: 2 x 5
  term         estimate std.error statistic  p.value
  <chr>           <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  1281120.    94527.     13.6  2.26e-30
2 roe_centered   18501.    11123.      1.66 9.78e- 2
```

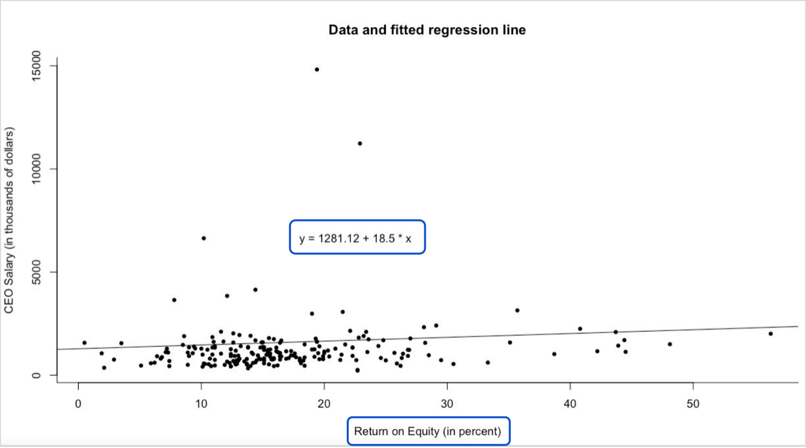
Changing units of measurement: Predictor

Here is a regression of CEO salary (in thousands of dollars) on return on equity (in percent).

If we divide or multiply the predictor variable by a constant *c*, then what happens to the intercept and slope estimates?

For example:
If we change return on equity (*in percent*) to
return on equity (*in decimal*) by dividing the variable by 100,
then what happens to the intercept and slope estimates?

(assuming nothing about the outcome variable has changed)

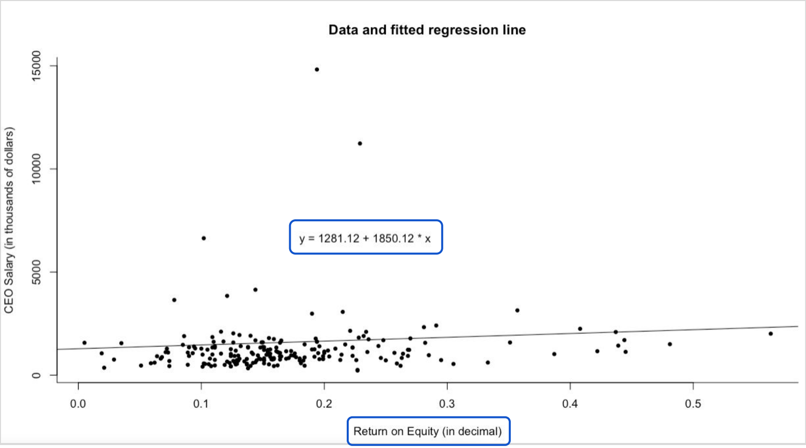



```
> # CEO Salary (in thousands of dollars)
> fit_ceosal1_ols <- lm(salary ~ roe_centered, data = ceosal1)
> tidy(fit_ceosal1_ols)
# A tibble: 2 x 5
  term         estimate std.error statistic  p.value
  <chr>           <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)     1281.      94.5     13.6  2.26e-30
2 roe_centered     18.5      11.1      1.66 9.78e- 2
```

Changing units of measurement: Predictor

If the predictor variable is **divided** or **multiplied** by a constant *c*, then the slope estimate is **multiplied** or **divided** by *c*, respectively.

Generally, **the intercept does not change** because it still corresponds to *f*(predictor) = 0




```
> # CEO Salary (in thousands of dollars) and ROE (in decimal)
> fit_ceosal1_ols_decimal <- lm(salary ~ roe_decimal, data = ceosal1)
> tidy(fit_ceosal1_ols_decimal)
# A tibble: 2 x 5
  term          estimate std.error statistic  p.value
  <chr>            <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)      1281.      94.5     13.6  2.26e-30
2 roe_decimal      1850.    1112.       1.66 9.78e- 2
```

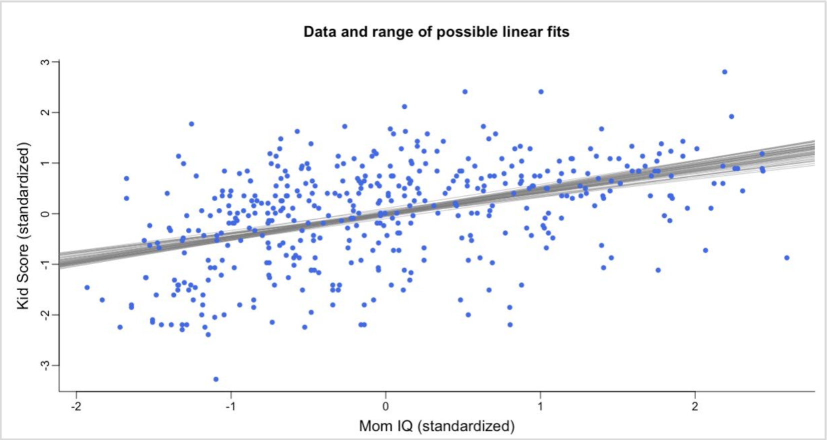
For example:
Original predictor: Return on equity (in percent)
Original intercept: 1281
Original coefficient: 18.50

Transformed predictor: Return on equity (in percent) **/ 100**
Untransformed intercept: 1281
Transformed coefficient: 18.50 *** 100** = 1850

Changing units of measurement: Standardize Outcome and Predictor

Here is a regression of kid score (standardized) on mom IQ (standardized).

How do we interpret the intercept and slope estimates?




```
stan_glm
 family:       gaussian [identity]
 formula:      kid_score_std ~ mom_iq_std
 observations: 434
 predictors:   2
------
            Median MAD_SD
(Intercept) 0.00   0.04
mom_iq_std  0.45   0.04
```

Changing units of measurement: Standardize Outcome and Predictor

The fitted model is:

kid_score_std = 0 + 0.45 * mom_iq_std + error

Standardized regression coefficients (b*) are defined as:

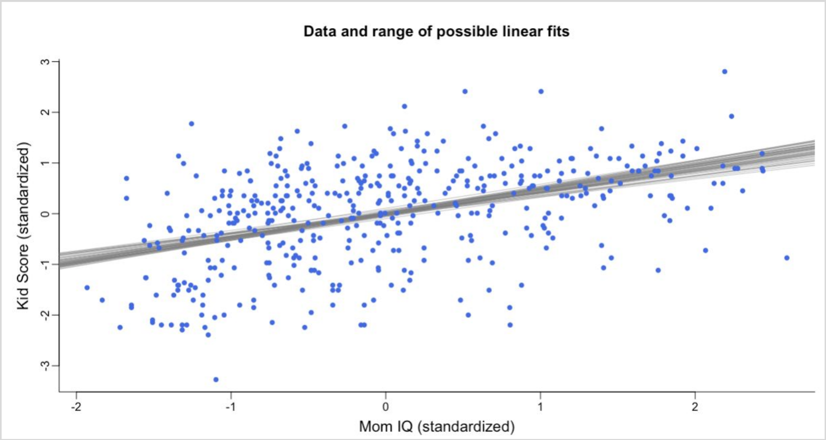
$\mathbf{b_k}^* = b_k * s_X / s_Y$

where b_k is the unstandardized regression coefficient, s_X is the standard deviation of X, and s_Y is the standard deviation of Y.

In standardized regression equations:

1. The Y-intercept always equals zero
2. All variables are expressed as standard scores, measured in standard deviations from their means
3. Coefficients indicate by how many standard deviations Y_hat changes, comparing subpopulations that differ by 1-standard-deviation in X (other variables held constant)

Like correlations, standardized regression coefficients theoretically range from –1 to +1.




```
stan_glm
 family:       gaussian [identity]
 formula:      kid_score_std ~ mom_iq_std
 observations: 434
 predictors:   2
------
             Median MAD_SD
(Intercept) 0.00   0.04
mom_iq_std  0.45   0.04
```

Changing units of measurement: Standardize Outcome and Predictor

The fitted model is:

kid_score_std = 0 + 0.45 * mom_iq_std + error

Standardized regression coefficients (b*) are defined as:

$\mathbf{b_k}^* = b_k * s_X / s_Y$

where b_1 is the unstandardized regression coefficient, s_X is the standard deviation of X, and s_Y is the standard deviation of Y.

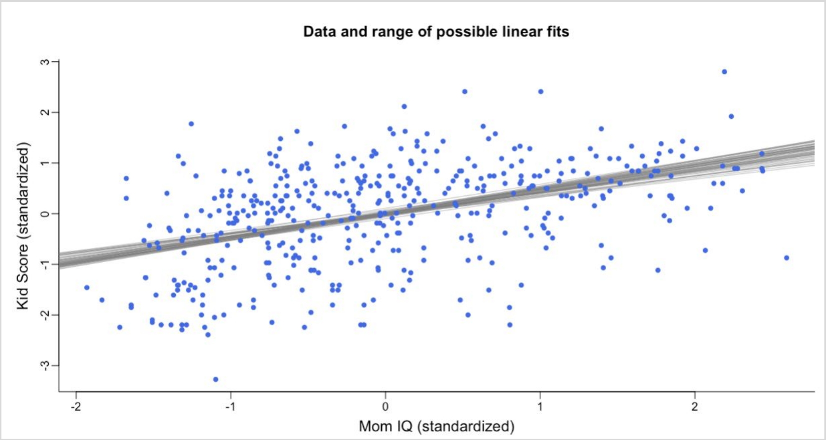
Interpretation:
If we compare subpopulations that differ in X by 1-standard-deviation, we expect the outcome for the group with higher X to be different by b_1^* standard deviations on average

For the kid score regression, $b_1 = 0.61$, $s_X = 15$, and $s_Y = 20.41$, so the standardized regression coefficient is:

$\mathbf{b_1}^* = 0.61 * 15 / 20.41$

$= 0.45$

That is, comparing kids whose moms differ in IQ by 1-standard-deviation (15 points), we expect the test scores for the kids whose mothers' IQs are higher to be greater by 0.45 standard deviations (about 0.45 * 20.41 = 9 points) on average




```
stan_glm
 family:       gaussian [identity]
 formula:      kid_score_std ~ mom_iq_std
 observations: 434
 predictors:   2
------
             Median MAD_SD
(Intercept) 0.00   0.04
mom_iq_std  0.45   0.04
```

Changing units of measurement: Power Transformations

Skew and outliers create problems even for simple statistics like the mean. They can cause issues for regression as well

Power transformations can reduce skew of univariate distributions:

- **q > 1**: Powers greater than 1 shift weight to the upper tail of the distribution and thereby *reduce negative skew*. The higher the power (2, 3, …), the stronger this effect
- **q = 1**: the raw data
- **q < 1**: Powers less than 1 pull in the upper tail and thereby *reduce positive skew*. The lower the power (.5, 0, –.5, …), the stronger this effect. To preserve order, add minus signs after raising to powers less than zero

Base 10 and base e logarithms have identical effects on distributional shape

By selecting an appropriate power transformation, we may be able to pull in outliers and make a skewed distribution more symmetrical (which can help mitigate statistical problems such as influence and heteroskedasticity)

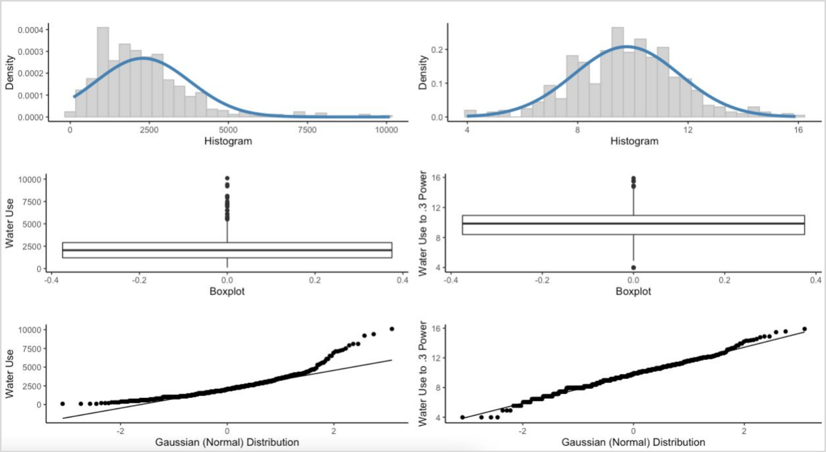
Ladder of Powers (Tukey, '77)	
Y^3	q = 3
Y^2	q = 2
Y^1	q = 1
$Y^{.5}$	q = .5
log Y	q = 0
$-(Y^{-.5})$	q = – .5
$-(Y^{-1})$	q = – 1

Changing units of measurement: Power Transformations

Power transformations of the outcome and predictor variables can reduce the skew of their univariate distributions, which can help mitigate statistical problems such as influence and heteroskedasticity

Transformation	Inverse Transformation	
$Y^* = Y^q$	$Y = Y^{*1/q}$	$q > 0$
$Y^* = \log_e Y$ or $Y^* = \log_{10} Y$	$Y = e^{Y^*}$ or $Y = 10^{Y^*}$	$q = 0$
$Y^* = -(Y^q)$	$Y = (-Y^*)^{1/q}$	$q < 0$

Concord Water Use:
As an example, in the dataset both *water use* and *income* are positively skewed. Let's use the ladder of powers to reduce their skew by raising both variables to the .3 power. The plots to the right show the distribution of *water use* before (left) and after (right) the transformation



With power transformed variables, how do we visualize and interpret the relationship between the predictor and outcome?

Changing units of measurement: Power Transformations

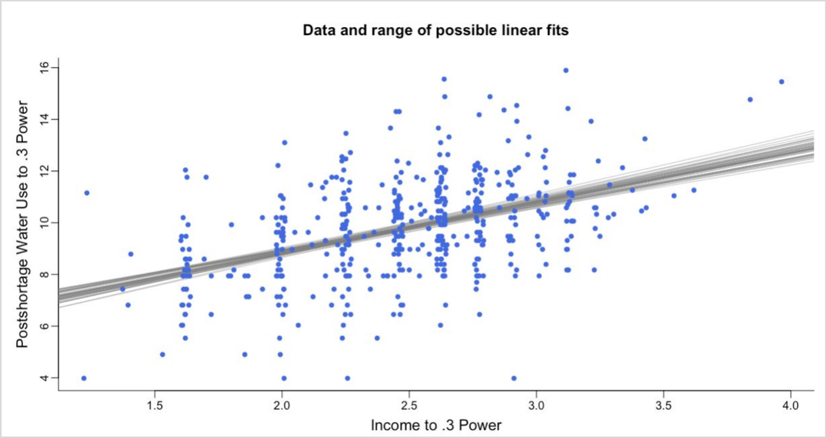
The fitted model is:

$$\text{Y\_hat}^* = 4.99 + 1.94 * X^* + \text{error}$$

$$\text{water\_use}^{.3} = 4.99 + 1.94 * \text{income}^{.3} + \text{error}$$

Interpreting points on fitted line:

This model asserts that the predicted .3 power of water use increases by 1.94 with every one-unit increase in the .3 power of income. *But what does this mean?*




```
stan_glm
 family:       gaussian [identity]
 formula:      water81_pt ~ income_pt
 observations: 496
 predictors:   2
------
             Median MAD_SD
(Intercept) 4.99   0.43
income_pt   1.94   0.17
```

Changing units of measurement: Power Transformations

Plots help us visualize the implications of transformed-variables regression:

First, obtain the predicted values (Y_hat*) from the transformed-variables equation. Then do the following:

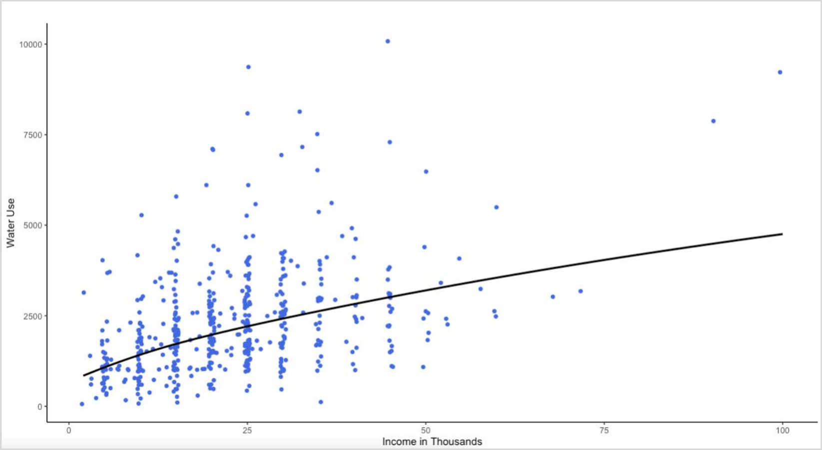
1. Convert transformed predicted values, Y_hat*, back into the natural units of Y (obtaining Y_hat)
2. Plot Y_hat against X

Perform step 1 only if Y was transformed. *If Y was not transformed, simply plot Y_hat against X*

For example, since we transformed water use (Y) by raising it to the .3 power ($Y^* = Y^{.3}$) the appropriate inverse transformation, applied to the predicted .3 power of water use (Y_hat*) is:

1. Y_hat = (Y_hat*)$^{1/.3}$
2. Plot Y_hat (predicted water use) against X (income)

Transformation	Inverse Transformation	
$Y^* = Y^q$	$Y = Y^{*1/q}$	$q > 0$
$Y^* = \log_e Y$ or $Y^* = \log_{10} Y$	$Y = e^{Y^*}$ or $Y = 10^{Y^*}$	$q = 0$
$Y^* = -(Y^q)$	$Y = (-Y^*)^{1/q}$	$q < 0$



Changing units of measurement: Transformations involving Logarithms

In applied work, you will encounter regression equations in which the outcome variable appears in logarithmic form. *Why is this done?*

Recall that in a simple linear regression, the coefficient for the slope denotes the *constant* difference in the outcome variable across the range of the predictor variable

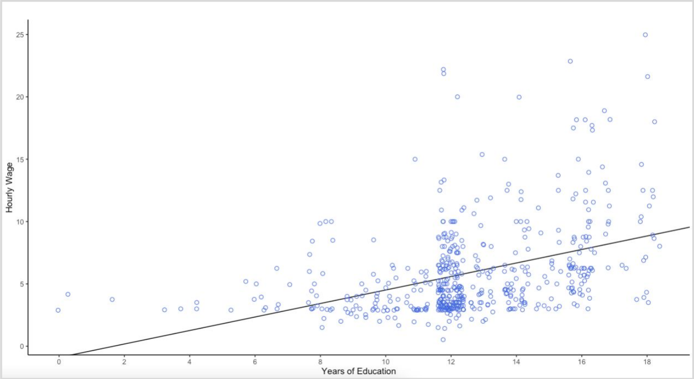
For example, in a regression of hourly wage on years of education, we obtain a slope estimate of 0.54, which means that comparing people who differ by 1 year of education we expect their hourly wage to differ by 54 cents on average (across all years of education)

For the simple linear regression, 54 cents is the difference between the 1st and 2nd years of education and the 17th and 18th years of education; *which may not be reasonable*.

Probably a better characterization of how wage changes with education is that each year of education increases wage by a constant *percentage*.

For example, the difference between the 1st and 2nd years of education is, say, 8%, and the difference between the 17th and 18th years of education is 8%

Model	**Outcome**	**Predictor**	**Interpretation of b_1**
Level-level	y	x	$\Delta y = b_1 \Delta x$
Level-log	y	log(x)	$\Delta y = (b_1/100)\%\Delta x$
Log-level	log(y)	x	$\%\Delta y = (100 * b_1)\Delta x$
Log-log	log(y)	log(x)	$\%\Delta y = b_1 \%\Delta x$



Changing units of measurement: Transformations involving Logarithms

Probably a better characterization of how wage changes with education is that each year of education increases wage by a constant *percentage*.

For example, the difference between the 1st and 2nd years of education is, say, 8%, and the difference between the 17th and 18th years of education is 8%

A model that gives (approximately) a constant percentage interpretation is:

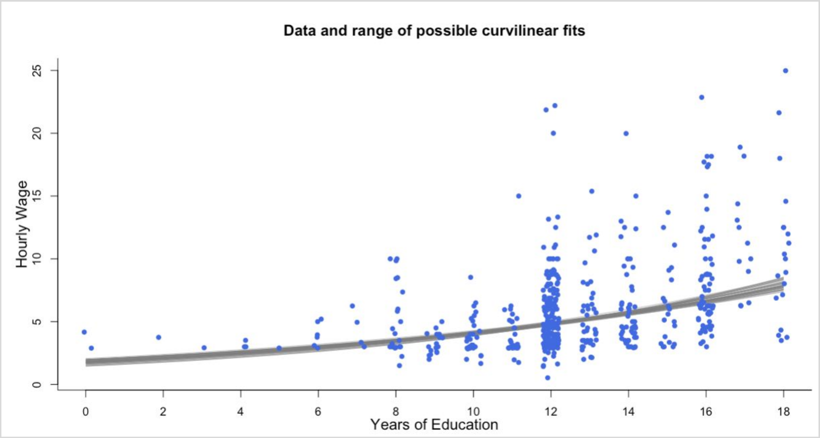
$$\log(wage) = a + b*educ + error$$

then

$$\%\Delta wage \cong (100 * b)\ \Delta educ$$

Because the percentage change in *wage* is the same for each additional year of education, the change in *wage* for an extra year of education *increases* as education increases

The coefficient on *educ* has a percentage interpretation when it is multiplied by 100: comparing people who differ by 1 year of education we expect their hourly wage to differ by 8.3% on average (across all years of education)




```
stan_glm
 family:       gaussian [identity]
 formula:      log(wage) ~ educ
 observations: 526
 predictors:   2
------
            Median MAD_SD
(Intercept) 0.585  0.097
educ        0.083  0.008
```

Changing units of measurement: Transformations involving Logarithms

Another model that gives (approximately) a constant percentage interpretation is a ***constant elasticity model***, which involves the natural logarithm of both the outcome and predictor:

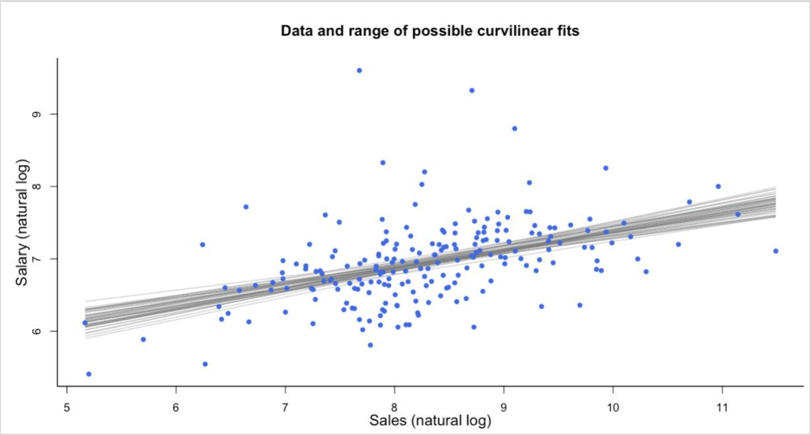
log(salary) = a + b*log(sales) + error

then

%Δsalary ≅ b * %Δsales

Interpretation:

The coefficient on *sales* has a percentage interpretation: comparing CEOs whose companies differ by 1% in sales we expect their salaries to differ by 0.257% on average




```
stan_glm
 family:       gaussian [identity]
 formula:      log(salary) ~ log(sales)
 observations: 209
 predictors:   2
------
             Median MAD_SD
(Intercept) 4.823  0.283
log(sales)  0.257  0.034
```

Appendix

Resources

Regression and Other Stories

Statistical Rethinking

Statistical rethinking with brms, ggplot2, and the tidyverse: Second edition

Bayes Rules!

Tidy Modeling with R

Doing Bayesian Data Analysis, Second edition

Doing Bayesian Data Analysis in brms and the tidyverse

rstanarm vignettes

bayesplot vignettes

R for Data Science

R Graphics Cookbook

Normal Distributions

Linear regression

Recall our linear model (kid_score = a + b*mom_iq + error), described with the following model components:

$y_i \sim \text{Normal}(\mu_i, \sigma)$ [likelihood]

$\mu_i = a + b(x_i - \text{mean}(x))$ [linear model]

$a \sim \text{Normal}(87, 20)$ [a prior]

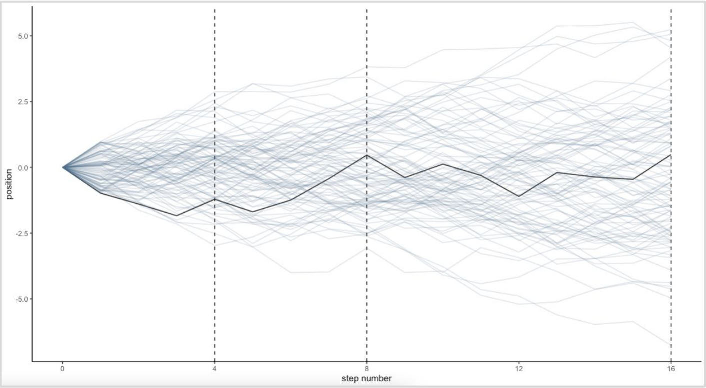
$b \sim \text{Normal}(0, 10)$ [b prior]

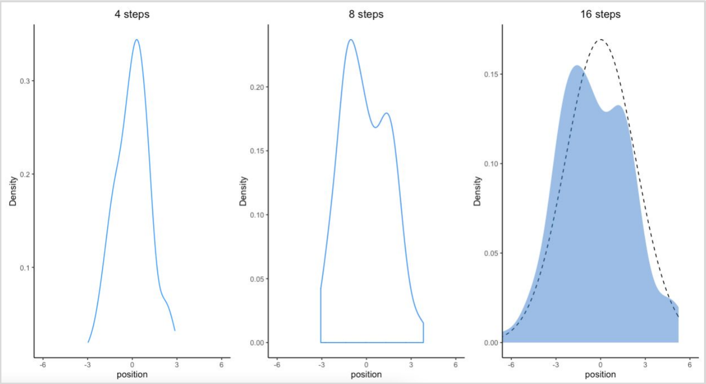
$\sigma \sim \text{Exponential}(1)$ [σ prior]

Why are normal distributions normal?

- Normal by addition
  - Any process that add together random values from the same distribution converges to a normal
  -
```r
# Normal by addition
pos <- replicate( 1000 , sum( runif(16 , -1 , 1) ) )
hist(pos)
plot(density(pos), xlab = "", main = "Normal by addition")
```
- Normal by multiplication
  - Any process that multiplies small deviations together tends to converge to a normal because multiplying small numbers is approximately the same as addition
  -
```r
# Normal by multiplication
growth <- replicate( 10000 , prod( 1 + runif(12 , 0 , 0.1) ) )
hist(growth)
plot(density(growth), xlab = "", main = "Normal by multiplication")
```
- Normal by log-multiplication
  - Any process that multiplies large deviations together tends to converge to a normal when measured on the log scale because adding logs is equivalent to multiplying the original numbers
  -
```r
# Normal by log-multiplication
log.big <- replicate( 10000 , log( prod( 1 + runif(12 , 0 , 0.5) ) ) )
hist(log.big)
plot(density(log.big), xlab = "", main = "Normal by log-multiplication")
```

Since measurement scales are arbitrary, all of these methods are legitimate





Why use normal distributions?

Two justifications for using the Gaussian distribution:

- Ontological (existence)
  - The world is full of Gaussian distributions, approximately
  - It is a widespread pattern at different scales and in different domains
  - Measurement errors, variations in growth, etc. tend towards Gaussian distributions because, at their heart, they add together fluctuations
  - There are many other patterns in nature; the Gaussian is a member of a family of fundamental natural distributions known as the exponential family

- Epistemological (state of knowledge)
  - The Gaussian represents a particular state of ignorance
  - When all we know about a distribution of measures is their mean and variance, then the Gaussian arises as the most consistent with our assumptions
  - If all we're willing to assume is that a measure has finite variance, then the Gaussian is the shape that can be realized in the largest number of ways (it is the least surprising and least informative assumption to make)