Lecture 4- Multiple Linear Regression

Peter Shaffery

1/26/2020

Staying Tidy

Last week we saw a package called `ggplot2`, which expands R's plotting capabilities. `ggplot2` is actually one package in a whole constellation called the `tidyverse`. The `tidyverse` is a (very large) collection of R packages meant to "rebuild" some of R's basic functionality, based around a principle called "tidy data".

I'm not going to go too deep into tidy data right now, but the basic idea is that each column in a dataset should correspond to a single variable, for example:

Untidy Data

year_2010	year_2011	year_2012
10	14	2
13	9	5
8	12	4

Tidy Data

year	counts
2010	10
2010	13
2010	8
2011	14
2011	9
2011	12
2012	2
2012	5
2012	4

Tidy data makes data manipulation a lot easier, and packages in the `tidyverse` take advantage of this to project some very powerful data manipulation tools. We've already seen one `tidyverse` function: `tidyr::drop_na`.

```
library(tidyverse) # imports a set of the "basic" tidyverse packages (including tidyr and dplyr)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(magrittr) # imports a code pipe %>%
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

One tool which makes working in the `tidyverse` is the **code pipe**: `%>%`, which comes from the package `magrittr`. For those of you familiar with Unix-derived OSs, this symbol operates similarly to the Unix `|`

```
x = 1:10
y = rnorm(length(x),mean=x)
mean(x)
```

```
## [1] 5.5
```

```
x %>% mean
```

```
## [1] 5.5
```

```
x %>% mean %>% cos
```

```
## [1] 0.7086698
```

```
cov(x,y)
```

```
## [1] 8.4037
```

```
x %>% cov(y)
```

```
## [1] 8.4037
```

```
x %>% cov(.,y)
```

```
## [1] 8.4037
```

One particularly handy variant of the standard code pipe, is the “assignment pipe”: `%<>%`

```
z = x
z %<>% mean
z
```

```
## [1] 5.5
```

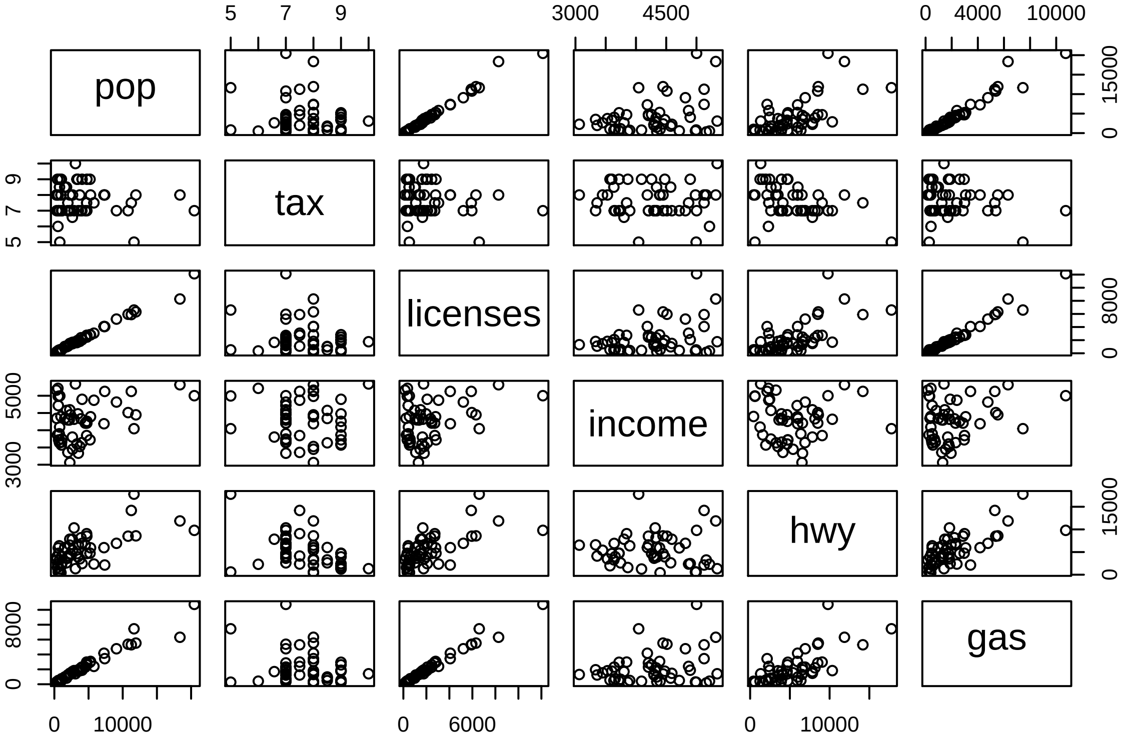
Fuel Analysis

We’ve been contracted by the US Department of Energy (DOE) to perform a study of the relationship between fuel taxes, and gasoline usage of automobiles.

```
# The dataset below contains data for the 50 states, including
# 1971 population in thousands (pop),
# 1972 gasoline tax in cents per gallon (tax),
# thousands of licensed drivers 1971 (licenses),
# per capita income in dollars (income),
# thousands of miles of federally funded highways in 1971 (hwy)
# total 1972 fuel consumption in millions of gallons (gas)

fuel = read.csv('../../data/fuel.csv')
fuel %<>% drop_na

fuel %>% head
```

```
##    pop  tax licenses income  hwy  gas state
## 1 1029  9.0      540   3571 1976  557    ME
## 2  771  9.0      441   4092 1250  404    NH
## 3  462  9.0      268   3865 1586  259    VT
## 4 5787  7.5     3060   4870 2351 2396    MA
## 5  968  8.0      527   4399  431  397    RI
## 6 3082 10.0     1760   5342 1333 1408    CT
```

```
fuel %>% dplyr::select(-state) %>% pairs
```



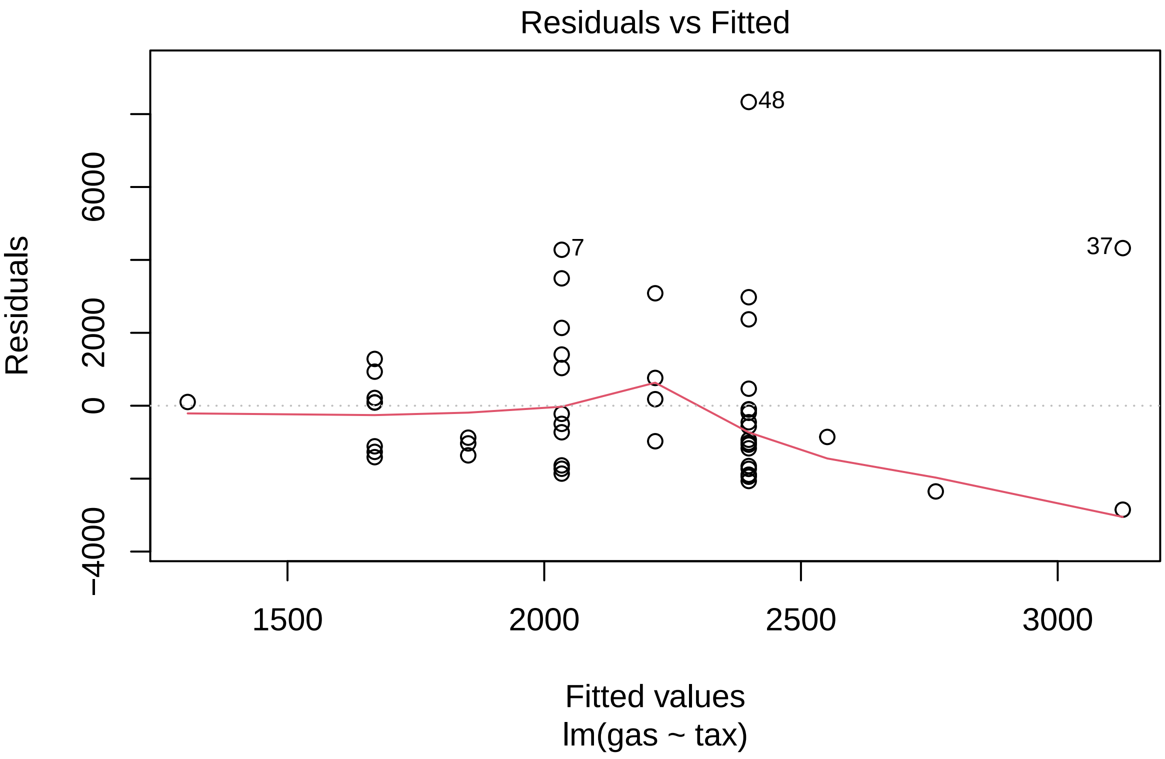
Let's start with a simple linear regression (SLR):

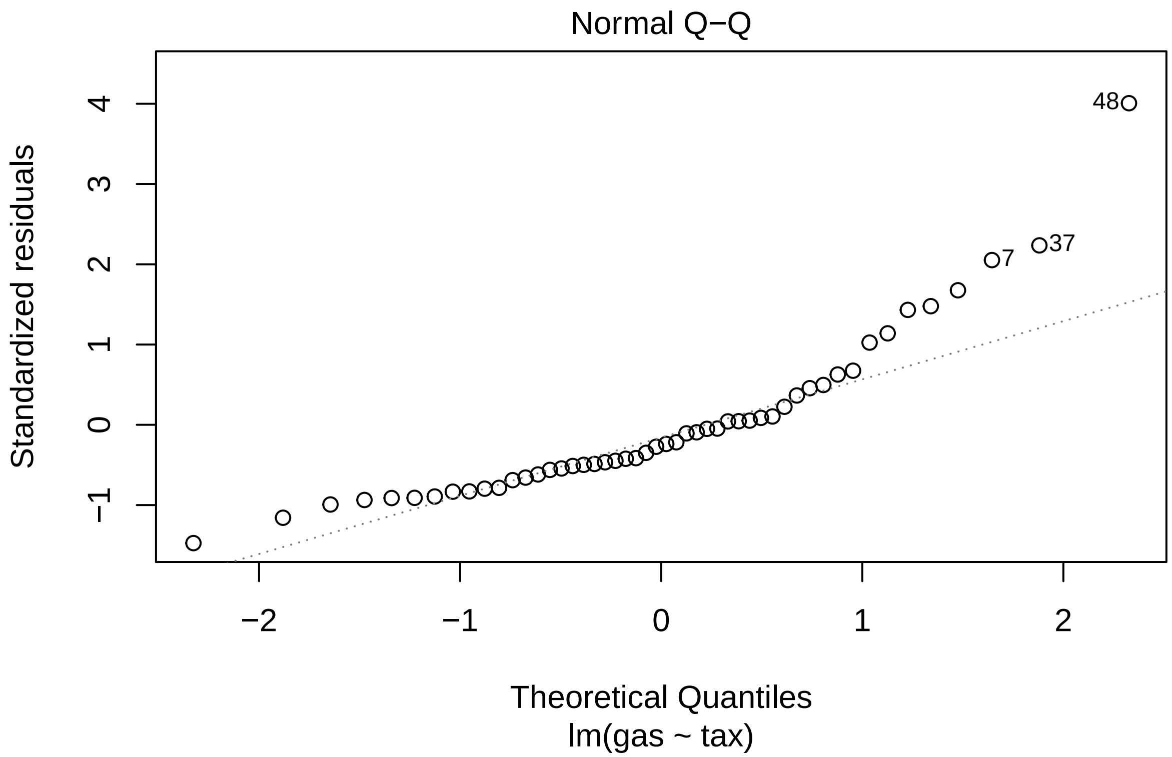
```
slr = lm(gas~tax, data=fuel)

summary(slr)
```

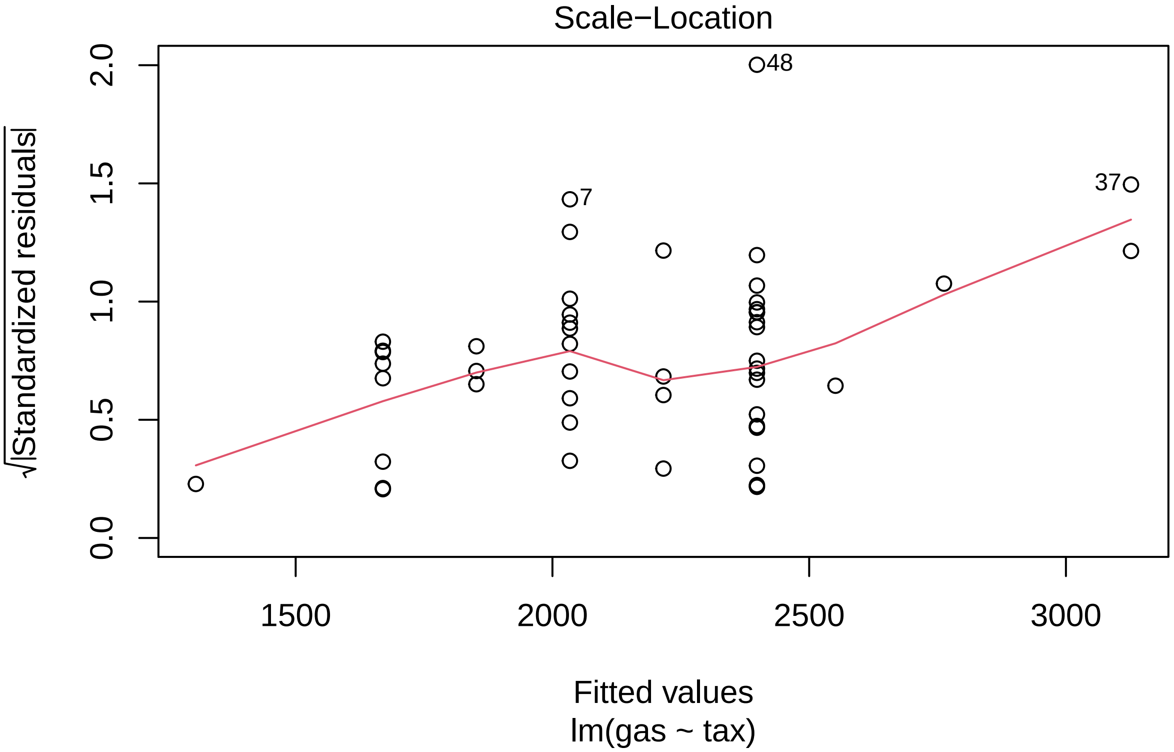
```
##
## Call:
## lm(formula = gas ~ tax, data = fuel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2850.7 -1337.9  -532.7   688.8  8331.7
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4947.9     2301.2   2.150   0.0366 *
## tax           -364.2      299.4  -1.217   0.2297
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2109 on 48 degrees of freedom
## Multiple R-squared:  0.02991,    Adjusted R-squared:  0.009704
## F-statistic:  1.48 on 1 and 48 DF,  p-value: 0.2297
```

```
plot(slr,which=c(1,2,3))
```







What do we think about this output? How do we interpret the things we're seeing here?

Explaining More Variability

Fuel consumption probably depends on more than just the fuel tax. How can we incorporate this information into our model?

Recall how we broke down the components of our SLR model:

$$y_i = (\beta_0 + \beta_1 x_i) + (\epsilon_i)$$
$$\text{Dep. Variable} = (\text{The stuff we explain}) + (\text{The stuff we can't explain})$$

In this example, our SLR model might be described as:

$$\text{GAS} = \beta_0 + \beta_1 \text{TAX} + \text{EVERYTHING ELSE}$$

Since EVERYTHING ELSE (by definition) depends on eg. LICENCES, we might write:

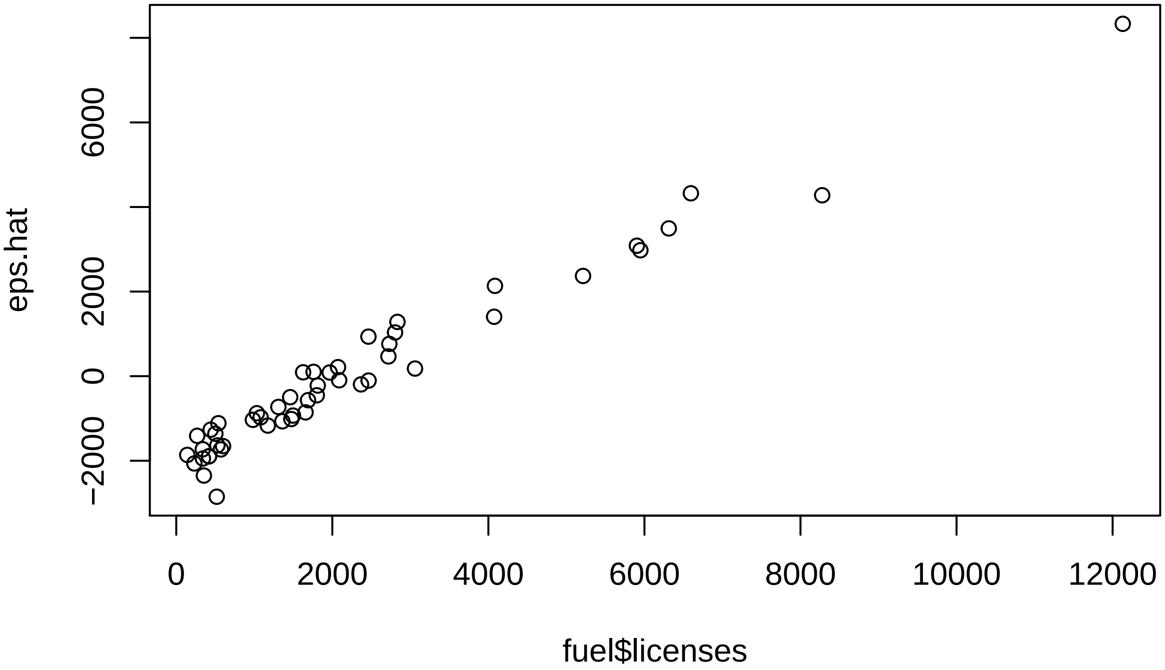
$$\text{EVERYTHING ELSE} = \beta_2 + \beta_3 \text{LICENSES} + \text{EVERYTHING BUT TAX AND LICENSES}$$

That is, we can model the error term **through a second SLR model**.

But how do we choose $\hat{\beta}_2$ and $\hat{\beta}_3$? For $\hat{\beta}_0$ and $\hat{\beta}_1$ we used our observations (x_i, y_i), but the ϵ_i are unobserved.

Well, recall that we have *estimates* of the ϵ_i, the residuals $\hat{\epsilon_i} = y_i - \hat{y_i}$. One option to fit $\hat{\beta}_2$ and $\hat{\beta}_3$ would be to run a SLR on the points $(z_i, \hat{\epsilon_i})$ where z_i is that i^{th} observation for the LICENSES variable:

```
eps.hat = resid(slr)
plot(fuel$licenses,eps.hat)
```



```
resid.mod.licenses = lm(eps.hat~fuel$licenses)
summary(resid.mod.licenses)
```

```
##
## Call:
## lm(formula = eps.hat ~ fuel$licenses)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1330.43  -261.85   -39.57   294.16   804.94
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.968e+03  8.242e+01  -23.88   <2e-16 ***
## fuel$licenses  8.629e-01  2.519e-02   34.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 417.9 on 48 degrees of freedom
## Multiple R-squared:  0.9607, Adjusted R-squared:  0.9599
## F-statistic:  1174 on 1 and 48 DF,  p-value: < 2.2e-16
```

Well okay, we've fit $\hat{\beta}_2$ and $\hat{\beta}_3$, but how do we interpret them?

We can sub our model for EVERYTHING ELSE directly into our model for GAS:

$$\text{GAS} = \beta_0 + \beta_1\text{TAX} + \beta_2 + \beta_3\text{LICENSES} + \text{EVERYTHING BUT TAX AND LICENSES}$$

And collecting up the constants gives us:

$$\text{GAS} = (\beta_0 + \beta_2) + \beta_1\text{TAX} + \beta_3\text{LICENSES} + \text{EVERYTHING BUT TAX AND LICENSES}$$

Note that since both β_0 and β_2 are constants, we could just write them as a single constant, β_0'

$$\text{GAS} = \beta_0' + \beta_1\text{TAX} + \beta_3\text{LICENSES} + \text{EVERYTHING BUT TAX AND LICENSES}$$

This a Multiple Linear Regression model (MLR). In order to clean up the notation a little, going forward I'll write this as:

$$\text{GAS}_i = \beta_0 + \beta_1\text{TAX}_i + \beta_2\text{LICENSES}_i + \epsilon_i$$

Or, more generally:

$$y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \epsilon_i$$

But *be careful*, I've overloaded (assigned multiple meanings to) each of the symbols here. The ϵ_i above represents EVERYTHING BUT TAX AND LICENSES, and **is not the same as the error term in our original SLR**.

In the SLR model, β_1 was interpreted as **the average change in y, for a unit change in x**. In the MLR model, the interpretation is similar, but with an important change.

Coefficient β_1 represents **the average change in y, for a unit change in x, at a fixed value of z**. That is, our MLR model *isolates* the effect of one independent variable on the dependent variable, *holding all other variables constant* (latin: *ceteris paribus*, "all else unchanged").

> If two states have the same number of licensed drives $z_i = z_j$, and differ in their fuel tax by a single unit $x_i - x_j = 1$, then *on average* their gas consumption will differ by $E[y_i - y_j] = \beta_1$. The interpretation of β_2 is similar

Fitting MLR in R

A closed form formula for $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2$ does exist (we'll see how to derive it next lecture). For now, however, let's see how R can be used to calculate $\hat{\beta}_1$ and $\hat{\beta}_2$ simultaneously:

```r
mlr = lm(gas~tax+licenses, data=fuel)
summary(mlr)
```

```
##
## Call:
## lm(formula = gas ~ tax + licenses, data = fuel)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1087.28   -99.33   -36.28   105.39  1240.76
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 902.04055  362.90691   2.486   0.0165 *
## tax         -96.04517   46.12069  -2.082   0.0428 *
## licenses      0.87770    0.01958  44.835   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 322.1 on 47 degrees of freedom
## Multiple R-squared:  0.9778, Adjusted R-squared:  0.9769
## F-statistic:  1037 on 2 and 47 DF,  p-value: < 2.2e-16
```

Let's compare the coefficients of the three models

```
slr
```

```
##
## Call:
## lm(formula = gas ~ tax, data = fuel)
##
## Coefficients:
## (Intercept)          tax
##      4947.9       -364.2
```

```
resid.mod.licenses # SLR on residuals
```

```
##
## Call:
## lm(formula = eps.hat ~ fuel$licenses)
##
## Coefficients:
##   (Intercept)  fuel$licenses
##    -1968.1838         0.8629
```

```
mlr
```

```
##
## Call:
## lm(formula = gas ~ tax + licenses, data = fuel)
##
## Coefficients:
## (Intercept)          tax     licenses
##    902.0406     -96.0452       0.8777
```

We see that the estimate for $\hat{\beta}_2$ is the same in both `mlr` and in `resid.mod.licenses`

Hypothesis Testing - MLR Style

Recall that in SLR we could perform procedure to select between the two possible hypotheses:

1. **Null Hypothesis (H_0):** $\beta_1 = 0$
2. **Alternate Hypothesis (H_A):** $\beta_1 \neq 0$

In MLR we test a similar pair of hypotheses:

1. **Null Hypothesis (H_0):** $\beta_1 = \beta_2 = 0$
2. **Alternate Hypothesis (H_A):** Either $\beta_1 \neq 0$ or $\beta_2 \neq 0$

Broadly, these hypotheses can be interpreted as measuring whether our entire model is any good.

To do so, we use a generalized variant of the *t-statistic*, called an **F-statistic**. In SLR the the F-statistic is (basically) the square of the t-statistic, so it doesn't matter which we use for hypothesis testing. For multi-dimensional models, however, they are different

$$F = \frac{\text{SSR}/k}{SSE/(n-k-1)}$$

Here n is the number of observations and k is the number of independent variables (in the example $k = 2$), while SSR and SSE have basically the same definitions as in SLR,

$$\hat{y_i} = \hat{\beta}_0 + \hat{\beta}_1 \text{TAX}_i + \hat{\beta}_2 \text{LICENSES}_i$$

$$SSE = \sum_{i=1}^{n} \hat{\epsilon_i}^2 = \sum_{i=1}^{n} (y_i - \hat{y_i})^2$$

$$SSR = \sum_{i=1}^{n} (\hat{y_i} - \bar{y})^2$$

I'm not going to go too much further with F-tests (if you're familiar with ANOVA, it is the same as here), but the core idea is that we can calculate a p-value with our F-statistic. R provides this as part of its standard regression table:

```r
# our whole model is not trash:
summary(mlr)
```

```
##
## Call:
## lm(formula = gas ~ tax + licenses, data = fuel)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1087.28   -99.33   -36.28   105.39  1240.76
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 902.04055  362.90691   2.486   0.0165 *
## tax         -96.04517   46.12069  -2.082   0.0428 *
## licenses      0.87770    0.01958  44.835   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 322.1 on 47 degrees of freedom
## Multiple R-squared:  0.9778, Adjusted R-squared:  0.9769
## F-statistic:  1037 on 2 and 47 DF,  p-value: < 2.2e-16
```

We see that since the p-value for this F-statistic is less than any reasonable α (such as $\alpha = .05$ or even $\alpha = .01$), that we can conclude that *either TAX or LICENSES has a significant effect on GAS.

Tangent: F-tests and p-hacking.

Recall that our alternate hypothesis stated that either $\beta_1 \neq 0$ or $\beta_2 \neq 0$. Let's pretend for a second that this alternate hypothesis was not true (ie. that we accepted its corresponding H_0). Based on this, we decide to add 10 more variables to the dataset, giving us coefficients $\beta_1, \beta_2, ..., \beta_{12}$.

Just by chance, there's a high probability that *at least one* of these 10 new variables has *some* correlation with y_i (recall that it's easy to find small correlations, even when there's no causal link).

The way that we've set up H_0 and H_A, we've made it very difficult for H_0 to fit the data. Briefly: H_A only has to get lucky once, H_0 **has to be lucky everytime**.

This asymmetry causes the p-value of an F-statistic to get very small as k (the number of predictors) grows large. Thus for even moderate values of k, the $F - statistic$ might be considered to have an outsized false positive rate. There are ways to account for this (such as the Bonferroni Correction), but my opinion this property makes F-tests more trouble than they're worth.

The practice of chucking more variables into your model in order to achieve significance is sometimes called "fishing", and is considered a type of p-hacking (techniques to manipulate a statistical procedure in order to get significance, and thereby score a publication or similar). P-hacking is considered unethical when done intentionally (you're effectively committing academic fraud), but can be easy to do by accident. Watch out for accidental p-hacking, lest you have to publish an embarassing retraction.

Hypothesis Testing - Individual Coefficients

You will notice that, in the regression table, R has assigned p-values to all the coefficients individually, not just the F-statistic for the whole model. How can we interpret these?

Recall that we can think about MLR as a chain of regressions, each time regressing the residuals on the covariates. The p-values of these regressions are what is shown in the regression table (note that, due to the way R calculates the MLR coefficients vs SLR, the actual numerical values are slightly different):

```
summary(resid.mod.licenses)
```

```
##
## Call:
## lm(formula = eps.hat ~ fuel$licenses)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1330.43  -261.85   -39.57   294.16   804.94
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.968e+03  8.242e+01  -23.88   <2e-16 ***
## fuel$licenses  8.629e-01  2.519e-02   34.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 417.9 on 48 degrees of freedom
## Multiple R-squared:  0.9607, Adjusted R-squared:  0.9599
## F-statistic:  1174 on 1 and 48 DF,  p-value: < 2.2e-16
```

```
summary(mlr)
```

```
##
## Call:
## lm(formula = gas ~ tax + licenses, data = fuel)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1087.28   -99.33   -36.28   105.39  1240.76
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 902.04055  362.90691   2.486   0.0165 *
## tax         -96.04517   46.12069  -2.082   0.0428 *
## licenses      0.87770    0.01958  44.835   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 322.1 on 47 degrees of freedom
## Multiple R-squared:  0.9778, Adjusted R-squared:  0.9769
## F-statistic:  1037 on 2 and 47 DF,  p-value: < 2.2e-16
```

That is (in this example), these p-values test between the following hypotheses:

- H_0: $\beta_{\text{LICENSES}} = 0$, given that we've included TAX in the model
- H_A: $\beta_{\text{LICENSES}} \neq 0$, given that we've included TAX in the model

Centering and Scaling

Looking coefficients in the regression table for our MLR, you might be tempted to conclude that TAX has a huge effect on GAS, relative to LICENSES. This is not the case, since these variables have *very different units* ("cents per gallon" vs "thousands of people"). This makes it hard to compare between the coefficients in our MLR.

One way that we can address this, is by converting all of our independent variables x_i to the same (unitless) scale. That is, for a single observation i of independent variable j, transforming it by:

$$x'_{j,i} = \frac{x_{j,i} - \bar{x}}{s_x}$$

Notice that now $E[x_{j,i}] = 0$ and $\text{Var}[x_{j,i}] = 1$. The $x'_{j,i}$ are now all in *units of standard deviation.*

Let's do this using R+tidyverse

```r
center.scale = function(x){
  x.bar = mean(x)
  s.x = sd(x)
  return((x-x.bar)/s.x)
}

# dplyr's `mutate` function adds new variables to a dataframe
fuel %<>% dplyr::mutate(
  tax.scaled=center.scale(tax),
  licenses.scaled=center.scale(licenses)
  )

mlr.scaled = lm(gas~tax.scaled+licenses.scaled, data=fuel)

summary(mlr)
```

```
##
## Call:
## lm(formula = gas ~ tax + licenses, data = fuel)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1087.28   -99.33   -36.28   105.39  1240.76
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 902.04055  362.90691   2.486   0.0165 *
## tax         -96.04517   46.12069  -2.082   0.0428 *
## licenses      0.87770    0.01958  44.835   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 322.1 on 47 degrees of freedom
## Multiple R-squared:  0.9778, Adjusted R-squared:  0.9769
## F-statistic:  1037 on 2 and 47 DF,  p-value: < 2.2e-16
```

```
summary(mlr.scaled)
```

```
##
## Call:
## lm(formula = gas ~ tax.scaled + licenses.scaled, data = fuel)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1087.28   -99.33   -36.28   105.39  1240.76
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2171.88      45.55  47.680   <2e-16 ***
## tax.scaled        -96.64      46.41  -2.082   0.0428 *
## licenses.scaled  2080.57      46.41  44.835   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 322.1 on 47 degrees of freedom
## Multiple R-squared:  0.9778, Adjusted R-squared:  0.9769
## F-statistic:  1037 on 2 and 47 DF,  p-value: < 2.2e-16
```

Looking at the scaled model we see that the effect of LICENSES is *two orders of magnitude* greater than the effect of TAX.