# Lecture 8

Peter Shaffery

2/9/2021

# Independent(?) Variables and Collinearity

## Dependent Independent Variables

When we first introduced the MLR model we showed how it could derived by regressing SLR residuals $\hat{\epsilon}_i$ on a new independent variable $x_2$. The idea here was that, by regressing the *unexplained residuals* against a new variable we were injecting additional information into our model to explain more overall variance in $y$.

One thing we *didn't* touch on in this explanation, is what happens if $x_1$ and $x_2$ have some relationship to each other?

The simplest way that $x_1$ and $x_2$ could be related is if $x_2 = x_1$. In this case we intuitively expect that adding $x_2$ into the regression shouldn't change anything, since by regressing $y$ on $x_1$ we have already "extracted" all the information out of $y$ that can be explained by $x_1$. Indeed, you can confirm that regressing $\hat{\epsilon}_i$ on $x_1$ gives a slope and intercept of 0:

```
library(tidyverse)
library(magrittr)
library(pracma)

fuel = read.csv('../../data/fuel.csv')
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
```

```
eps.hat = resid(lm(gas~tax,data=fuel))

x2 = fuel$tax
lm(eps.hat~x2)
```

```
##
## Call:
## lm(formula = eps.hat ~ x2)
##
## Coefficients:
## (Intercept)           x2
##  -6.431e-14    0.000e+00
```

```
x2 = 10*x2
lm(eps.hat~x2)
```

```
##
## Call:
## lm(formula = eps.hat ~ x2)
##
## Coefficients:
## (Intercept)           x2
##   1.166e-12   -1.614e-14
```

Now let's choose a slightly looser relationship between $x_1$ and $x_2$, let's say the following:

$$x_2 = \alpha_0 + \alpha_1 x_1 + \eta$$

Where $\eta$ is a random variable that accounts for all the variation in $x_2$ *not* explained by $x_1$.

Let's plug this into MLR model:

$$\begin{aligned}
y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon \\
y &= \beta_0 + \beta_1 x_1 + \beta_2(\alpha_0 + \alpha_1 x_1 + \eta) + \epsilon \\
y &= (\beta_0 + \beta_2\alpha_0) + (\beta_1 + \beta_2\alpha_1)x_1 + \beta_2\eta + \epsilon
\end{aligned}$$

...and relabeling terms:

$$\begin{aligned}
\beta_0' &= \beta_0 + \beta_2\alpha_0 \\
\beta_1' &= \beta_1 + \beta_2\alpha_1 \\
\epsilon' &= \beta_2\eta + \epsilon
\end{aligned}$$

We now see that we have gone *back* to our original SLR model:

$$y = \beta_0' + \beta_1' x_1 + \epsilon'$$

Where $\beta_0'$ and $\beta_1'$ are the SLR estimates obtained by regressing $y$ on $x_1$.

```
slr = lm(gas~tax,data=fuel)
mlr = lm(gas~tax+licenses,data=fuel)
slr.tax.lic = lm(licenses~tax,data=fuel)

beta1.prime = coef(slr)[2]
beta1 = coef(mlr)[2]
beta2 = coef(mlr)[3]
alpha1 = coef(slr.tax.lic)[2]

print(c(beta1+beta2*alpha1, beta1.prime))
```

```
##       tax       tax
## -364.2309 -364.2309
```

If we unwind this proof like one step, we come to a really neat interpretation of $\beta_2$:

$$\begin{aligned}
\epsilon' &= \beta_2\eta + \epsilon \\
y - \beta_0' - \beta_1' x_1 &= \beta_2(x_2 - \alpha_0 - \alpha_1 x_1) + \epsilon \\
y &= \beta_0' + \beta_1' x_1 + \beta_2\eta + \epsilon
\end{aligned}$$

Here you can think of $\eta = (x_2 - \alpha_0 - \alpha_1 x_1)$ as the residuals of $x_2$ regressed on $x_1$. More qualitatively, $\eta$ is the **new information that $x_2$ introduces to the model**, and $\beta_2$ captures how this *new information* influences $y$ (through $\epsilon$).

### Example: Fuel

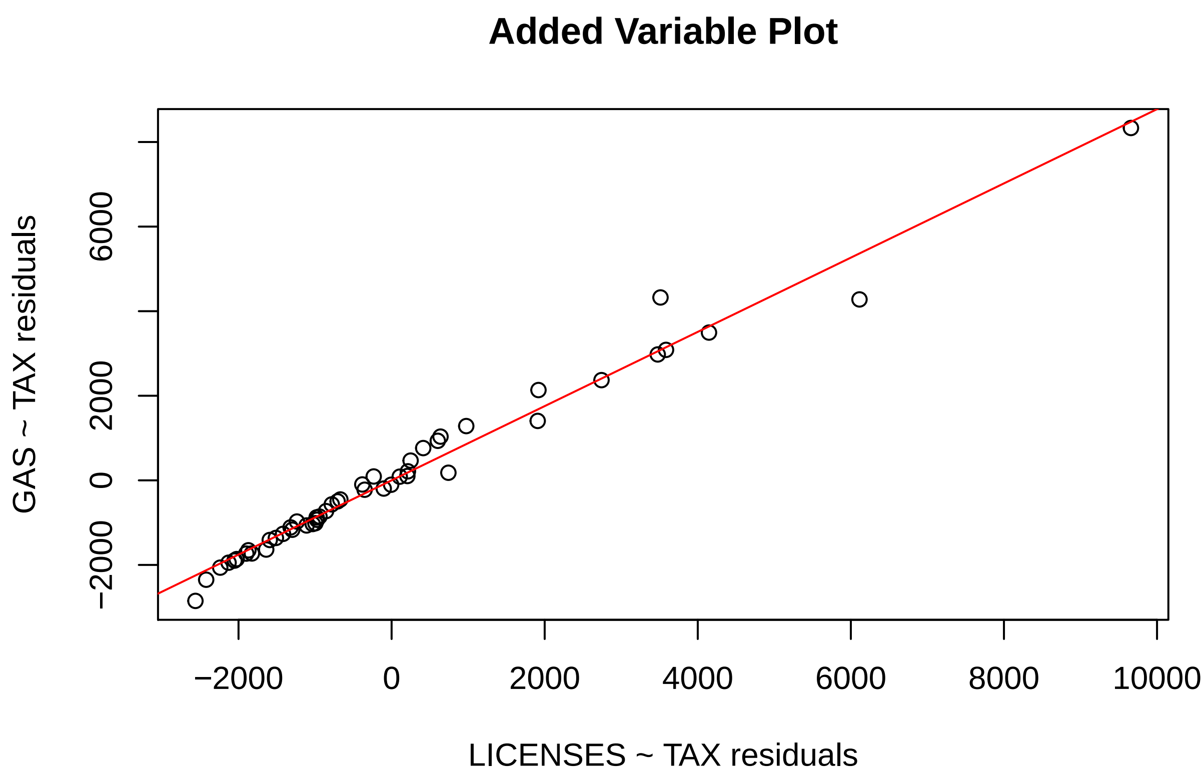
One way to assess how much new information an additional variable brings to a model is through **added variable plots**.

Conceptually, you can think about these as showing $\epsilon'$ vs. $\eta$.

More concretely: for a covariate $x_k$ being added to covariates $x_1, ..., x_{k-1}$ an added variable plot shows the residuals of $x_k$ regressed on $x_1, ..., x_{k-1}$ on the x-axis, and the residuals of $y$ regressed on $x_1, ..., x_{k-1}$ on the x-axis.

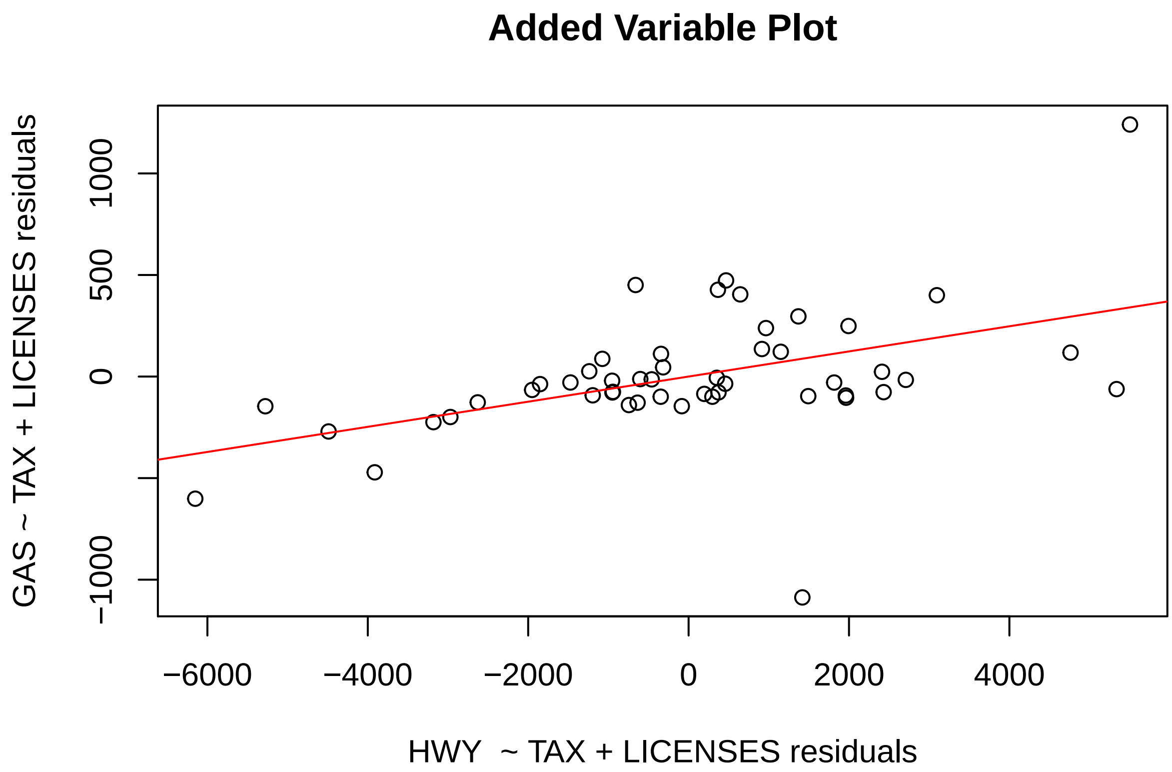
```
slr = lm(gas~tax, data=fuel)
av.licenses = lm(licenses~tax, data=fuel)

x.resid = resid(av.licenses)
y.resid = resid(slr)
plot(x.resid,y.resid,
     xlab='LICENSES ~ TAX residuals',
     ylab='GAS ~ TAX residuals',
     main='Added Variable Plot')
abline(lm(y.resid~x.resid), col='red')
```




```
mlr = lm(gas~tax + licenses, data=fuel)
av.hwy = lm(hwy~tax+licenses, data=fuel)

x.resid = resid(av.hwy)
y.resid = resid(mlr)
plot(x.resid,y.resid,
     xlab='HWY  ~ TAX + LICENSES residuals',
     ylab='GAS ~ TAX + LICENSES residuals',
     main='Added Variable Plot')
abline(lm(y.resid~x.resid), col='red')
```




## Covariance

Let's go back to example $x_2 = x_1$. As we discussed above, adding $x_2$ to the MLR model in this case will have no overall effect on this model. For a covariate to be useful *it needs to introduce new information.*

What about a case where $x_2$ introduces only a small amount of new information?

One way that we could quantify this case is with $Cov(x_1, x_2)$. Going back to our model:

$$x_2 = \alpha_0 + \alpha_1 x_1 + \eta$$

If $Cov(x_1, x_2)$ is relatively large, then $\alpha_1$ will dominate the effect of $\eta$, and so the novel information introduced by $x_2$ will be small.

Will this effect our regression results at all?

To answer that, we'll need to talk about **covariance matrices**.

Recall that, for example, a multivariate normally distributed vector $\vec{z}$ we needed to specify its variance matrix $\Sigma$. We had said that the diagonal elements of this matrix gave us the variance of the vector elements, $\Sigma_{ii} = \text{Var}[z_i]$.

As it turns out, the off-diagonal elements give us the *covariance* between the elements of this vector:

$$\Sigma_{ij} = \text{Cov}(z_i, z_j)$$

For the case of the covariates $x_1$ and $x_2$, it turns out that our covariance matrix has a familiar form. Let's recall the definition of sample covariance:

$$\text{Cov}[x_1, x_2] = \frac{1}{n-1} \sum_{i=1}^{n} (x_{1,i} - \bar{x}_1)(x_{2,i} - \bar{x}_2)$$

For the time-being let's assume that $E[x_j] = \bar{x}_j = 0$, so $\mathrm{Cov}[x_1, x_2] = \frac{1}{n-1} \sum_{i=1}^{n} x_{1,i} x_{2,i}$.

Looking at this, it's not hard to convince yourself that it's the $i$th, $j$th element of the matrix $X^T X$, where $X$ is defined:

$$X = \begin{bmatrix} x_{1,1} & x_{2,1} \\ \vdots & \vdots \\ x_{1,n} & x_{2,n} \end{bmatrix}$$

**What is the shape of $X^T X$ in this example?**

That matrix $X^T X$ we have seen in (among other places) the formula for $\hat{\beta} = (X^T X)^{-1} X^T y$. This should clue us in that the problem of tightly related variables can hit pretty deeply in the regression process.

Let's pretend, again, that $x_1 = x_2$. In this case we don't have to do any matrix multiplication to obtain $X^T X$. Since $\mathrm{Cov}[x_1, x_2] = \mathrm{Cov}[x_1, x_1] = \mathrm{Var}[x_1]$ we immediately see that the matrix will be given:

$$X^T X = \mathrm{Var}[x_1] \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

This is major problem, because it turns out that this matrix is not invertible:

```
library(pracma)
bad.matrix = ones(2,2)
inv(bad.matrix)
```

```
## Warning in inv(bad.matrix): Matrix appears to be singular.
```

```
##      [,1] [,2]
## [1,]  Inf  Inf
## [2,]  Inf  Inf
```

So in the case where $x_1 = x_2$ we can't even perform MLR (unless we get rid of $x_2$ and just go back to SLR).

In cases where $x_1$ and $x_2$ are not related, but *are* tightly correlated, then the matrix $X^T X$ will be **nearly non-invertible**. Such a matrix is sometimes said to be "ill-conditioned" (it will have a very high condition number, for those of you who have taken numerics).

This problem is referred to as **collinearity**, since from a linear algebra perspective it is equivalent to the columns of $X$ having a lot of linear dependence (ie. sharing a hyperplane). Collinearity can result in a number of undesirable consequences:

1. High sensitivity to data: similar to the case of high leverage, when the covariates are collinear the addition or removal of a small handful of data points can dramatically change the estimated coefficients
2. High parameter uncertainty: if $x_1$ and $x_2$ are tightly related, it is hard to separate out the effects of one from the other. This manifests in their coefficients having large confidence intervals, or equivalently, high p-values
3. Wacky parameter values: similar to 2, because the model has a hard time separating the effects of $x_1$ and $x_2$, the estimated coefficient values may be surprising (negative when they should be positive)

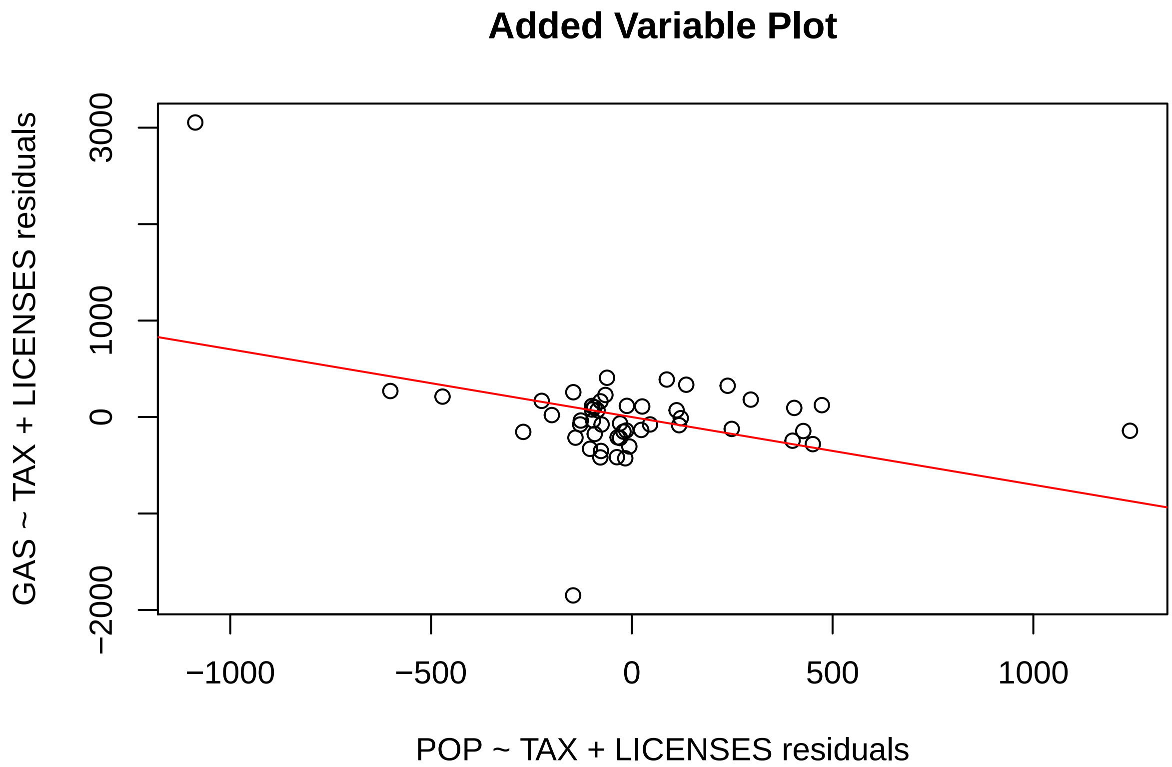
Because of these problems, it's important to identify collinear variables and remove them from your model (since the information they include is mostly already in there anyways!)

```
mlr = lm(gas~tax+licenses, data=fuel)
av.pop = lm(pop~tax+licenses, data=fuel)

x.resid = resid(mlr)
y.resid = resid(av.pop)

plot(x.resid,y.resid,
```

```
     xlab='POP ~ TAX + LICENSES residuals',
     ylab='GAS ~ TAX + LICENSES residuals',
     main='Added Variable Plot')
abline(lm(y.resid~x.resid), col='red')
```




```
mlr %>% summary
```

```
##
## 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
```

```
lm(gas~tax+licenses+pop, data=fuel) %>% summary
```

```
##
```

```
## Call:
## lm(formula = gas ~ tax + licenses + pop, data = fuel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -567.46 -134.57  -69.68  114.09 1208.31
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 651.40020  346.70830   1.879  0.06662 .
## tax         -65.54387   43.95275  -1.491  0.14273
## licenses      1.29993    0.14380   9.040 9.11e-12 ***
## pop          -0.22783    0.07697  -2.960  0.00485 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 298.4 on 46 degrees of freedom
## Multiple R-squared:  0.9814, Adjusted R-squared:  0.9802
## F-statistic: 808.2 on 3 and 46 DF,  p-value: < 2.2e-16
```

# Diagnosing Collinearity

## Covariance and Correlation Matrix

We've already talked about covariance matrices, and correlation matrices are defined similarly (the matrix whos $i$th $j$th matrix is $\mathrm{Cor}(x_i, x_j)$). R provides these functions natively:

```
cov(fuel %>% select(-state))
```

```
##                    pop          tax      licenses       income          hwy
## pop      19470466.0669  -437.750400 10372283.9633 878203.00408 10133990.009
## tax         -437.7504     1.012389     -309.3409    -26.56457    -1414.886
## licenses 10372283.9633  -309.340898  5619172.8571 462760.89796  5455909.706
## income     878203.0041   -26.564571   462760.8980 343327.43878   -18507.498
## hwy      10133990.0090 -1414.886139  5455909.7061 -18507.49796 12283952.104
## gas       9075916.8718  -368.743478  4961656.5469 322027.91429  5279533.134
##                  gas
## pop      9075916.8718
## tax         -368.7435
## licenses 4961656.5469
## income    322027.9143
## hwy      5279533.1339
## gas      4489770.7200
```

```
cor(fuel %>% select(-state))
```

```
##                 pop         tax   licenses       income          hwy
## pop       1.00000000 -0.09859721  0.9916308  0.339666321  0.655274459
## tax      -0.09859721  1.00000000 -0.1296962 -0.045058300 -0.401216401
## licenses  0.99163085 -0.12969624  1.0000000  0.333170240  0.656692295
## income    0.33966632 -0.04505830  0.3331702  1.000000000 -0.009012067
## hwy       0.65527446 -0.40121640  0.6566923 -0.009012067  1.000000000
## gas       0.97071168 -0.17295715  0.9878214  0.259374443  0.710909603
##                gas
## pop      0.9707117
```

```
## tax      -0.1729571
## licenses  0.9878214
## income    0.2593744
## hwy       0.7109096
## gas       1.0000000
```

### Pairwise Scatterplots

While covariance and correlation are useful numerically, they don't always tell the whole story. Say that:

$$x_2 = x_1^2$$

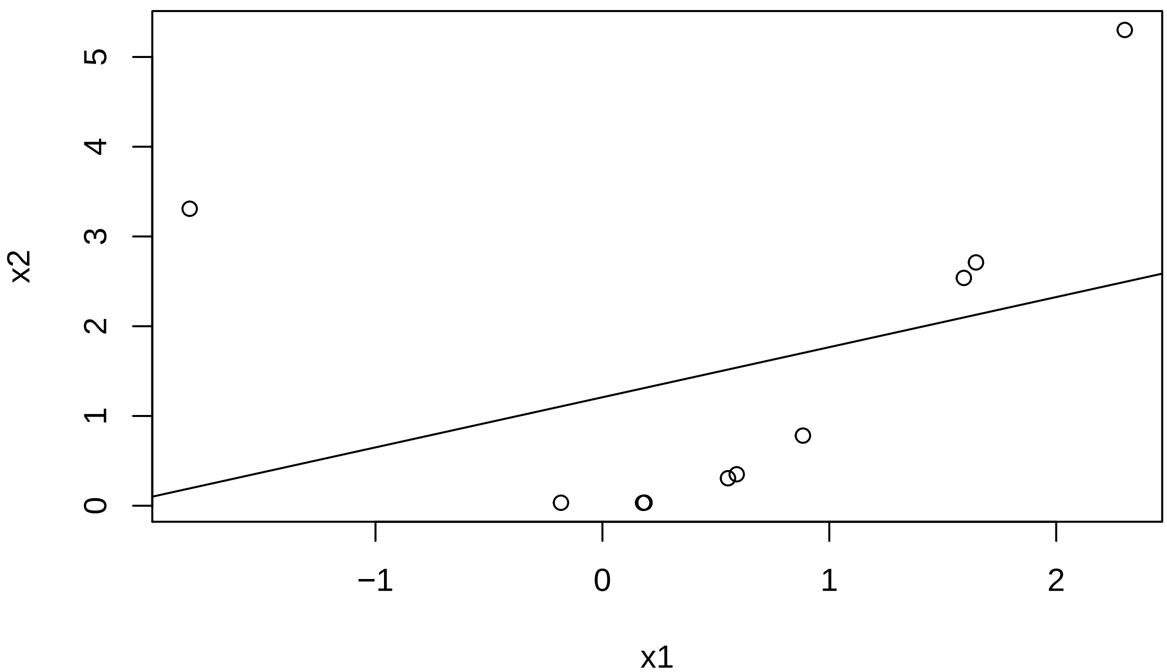
In this case the correlation can be fairly low, even though $x_2$ is a direct function of $x_1$:

```
set.seed(22345)
x1 = rnorm(10)
x2 = x1^2

cor(x1,x2)
```

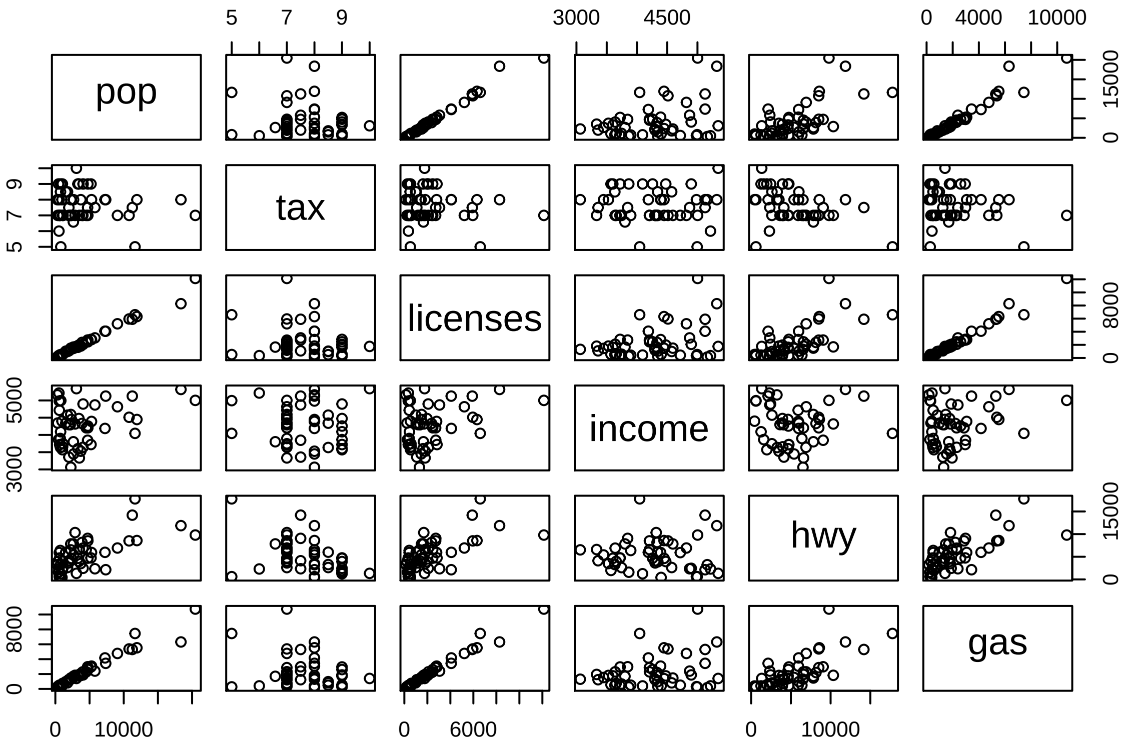
```
## [1] 0.3514439
```

```
plot(x1,x2)
abline(lm(x2~x1))
```




Therefore it is always important to supplement a covariance matrix with a **pairwise scatterplot** (or "scatterplot matrix"). This will help catch cases where a non-linear relationship may fool a covariance matrix approach:

```r
pairs(fuel %>% select(-state))
```




## Variance Inflation Factor (VIF)

A problem with both of the two prior methods is that they only assess *pairwise* collinearity (ie. between LICENSES and POP). However often it is the case that one variable is collinear with a whole collection of other variables, rather than any one variable in particular:

$$x_k = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_{k-1} x_{k-1}$$

In this case our pairwise diagnostics will fail us.

One way that we could diagnose this is with added variable plots, since here the residuals of $x_k$ regressed on $x_1, ..., x_{k-1}$ will not be informative of the residuals of $y$ regressed on $x_1, ..., x_{k-1}$.

However it could also be the case that $x_k$ is not collinear with $x_1, ..., x_{k-1}$, but is *also* not informative of $y$, and in this case the AV plot will show the same thing.

We therefore turn to the **Variance Inflation Factor**, which is defined for each variable in our model:

$$V_k = \frac{1}{1 - R_k^2}$$

Where $R_k^2$ is the $R^2$ obtained from regression $x_k$ on $x_1, ..., x_{k-1}$.

Since $R^2$ is bounded in $[0, 1]$, the "best" VIF is $V_k = 1$, since here the new variable is perfectly unrelated to everything else in the model. In general any $V_k < 10$ is usually considered "good".

The name "variance inflation factor" comes from the fact that:

$$\text{Var}[\beta_j] \propto V_j$$

Driving home the point that collinearity increases uncertainty in our inference.

Computing VIF is typically done using a package (and a lot of options exist). Here we use the `car` package to compute the VIF for all possible variables in the `fuel` dataset:

```r
full.mod = lm(gas ~ ., data=fuel%>%select(-state))
car::vif(full.mod)
```

```
##       pop       tax  licenses    income       hwy 
## 66.589957  1.395434 64.281030  1.314848  2.551862
```