Added Variable Plots

Derek Sonderegger

```r
library(tidyverse)

# Install from Derek's GitHub just once. Afterwards you can load
# the library as normal.
# devtools::install_github('dereksonderegger/ggAVplots')
library(ggAVplots)
```

Theory

The Zagat guide contains restaurant ratings and reviews for many major world cities. We want to understand variation in the average Price of a dinner in Italian restaurants in New York City. Specifically, we want to know how customer ratings (measured on a scale of 0 to 30) of the Food, Decor, and Service, as well as whether the restaurant is located to the east or west of 5th Avenue, affect the average Price of a meal. The data contains ratings and prices for 168 Italian restaurants in 2001.

This material for this activity was adapted from Sheather, A Modern Approach to Regression with R by Amelia McNamara. I've added some updates.

```r
library(tidyverse)

nyc <- read.csv("http://www.math.smith.edu/~bbaumer/mth247/sheather/nyc.csv")
dim(nyc)
```

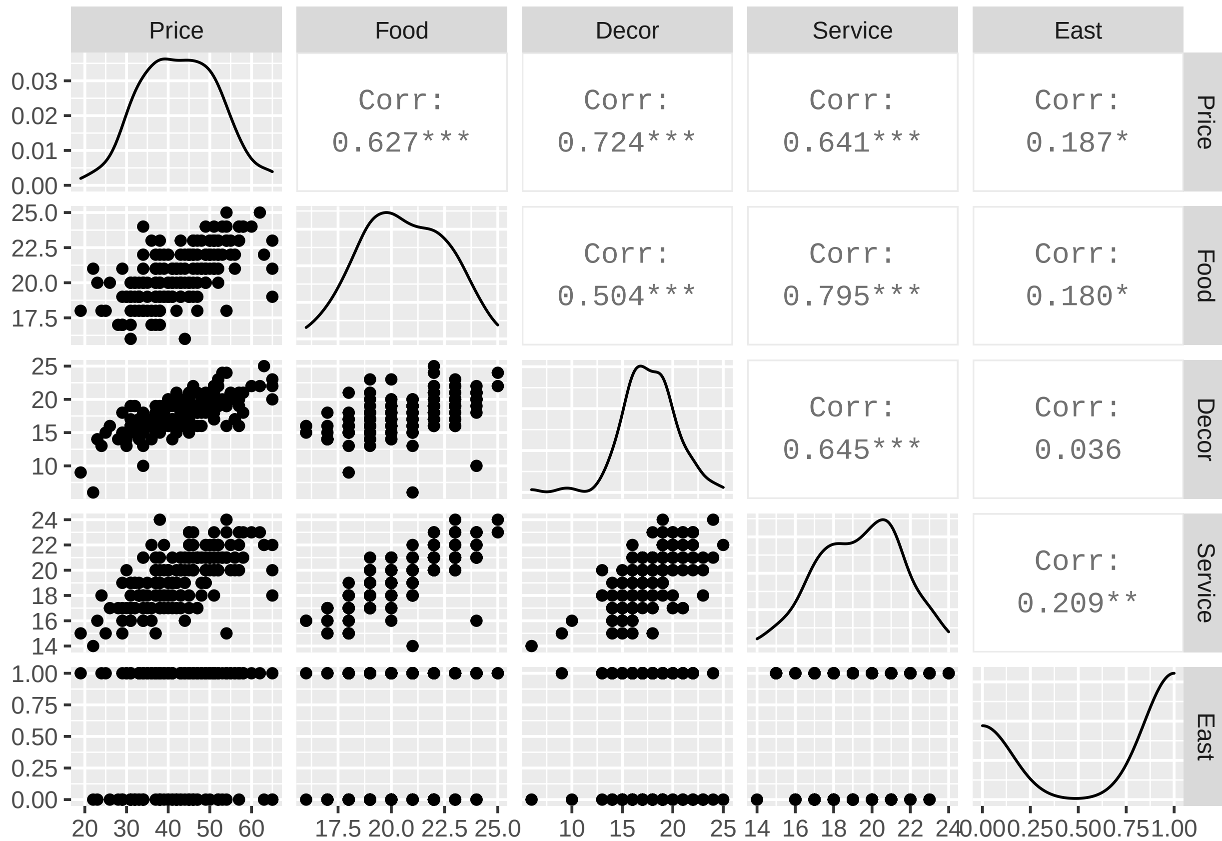
```
## [1] 168   7
```

```r
head(nyc)
```

```
##   Case          Restaurant Price Food Decor Service East
## 1    1 Daniella Ristorante    43   22    18      20    0
## 2    2  Tello's Ristorante    32   20    19      19    0
## 3    3          Biricchino    34   21    13      18    0
## 4    4             Bottino    41   20    20      17    0
## 5    5          Da Umberto    54   24    19      21    0
## 6    6            Le Madri    52   22    22      21    0
```

Lets check out the correlation plots first

```r
nyc %>%
  select(Price:East) %>%
  GGally::ggpairs()
```



Unsurprisingly, food, decor, and service all all highly correlated.

Questions

Which variables seems to be strongly correlated with Price?

Are there other significant relationships between the variables that seem important? Generate a correlation matrix to quantify relationships between individual pairs of variables.

```
nyc %>%
  select(Price:East) %>%
  cor() %>%
  round( digits=3 )
```

```
##         Price  Food Decor Service  East
## Price   1.000 0.627 0.724   0.641 0.187
## Food    0.627 1.000 0.504   0.795 0.180
## Decor   0.724 0.504 1.000   0.645 0.036
## Service 0.641 0.795 0.645   1.000 0.209
## East    0.187 0.180 0.036   0.209 1.000
```

Clearly food, decor, and service all are correlated with the price, but because they are correlated with each other, we have to be careful in interpeting the coefficients.

One way to understand the effect of, say service, after accounting for food and decor is something called an "added variable plot" or "partial regression plot".

If we first consider the full model with all the variables.

```r
m_full <- lm(Price ~ Food + Decor + Service + East, data=nyc)
summary(m_full)
```

```
##
## Call:
## lm(formula = Price ~ Food + Decor + Service + East, data = nyc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.0465  -3.8837   0.0373   3.3942  17.7491
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.023800   4.708359  -5.102 9.24e-07 ***
## Food          1.538120   0.368951   4.169 4.96e-05 ***
## Decor         1.910087   0.217005   8.802 1.87e-15 ***
## Service      -0.002727   0.396232  -0.007   0.9945
## East          2.068050   0.946739   2.184   0.0304 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.738 on 163 degrees of freedom
## Multiple R-squared:  0.6279, Adjusted R-squared:  0.6187
## F-statistic: 68.76 on 4 and 163 DF,  p-value: < 2.2e-16
```

These coefficients don't necessarily make sense to me. In particular I don't understand why `Decor` has such a strong p-value but `Service` has almost a negligible (but negative!) effect.

Added Variable Plot procedure

Consider the set of $k+1$ variables $X_1, X_2, \dots, X_k, Z$ where we are interested in the effect of Z on the response variable after accounting for the other $X_1, \dots, X_k$. The procedure is:

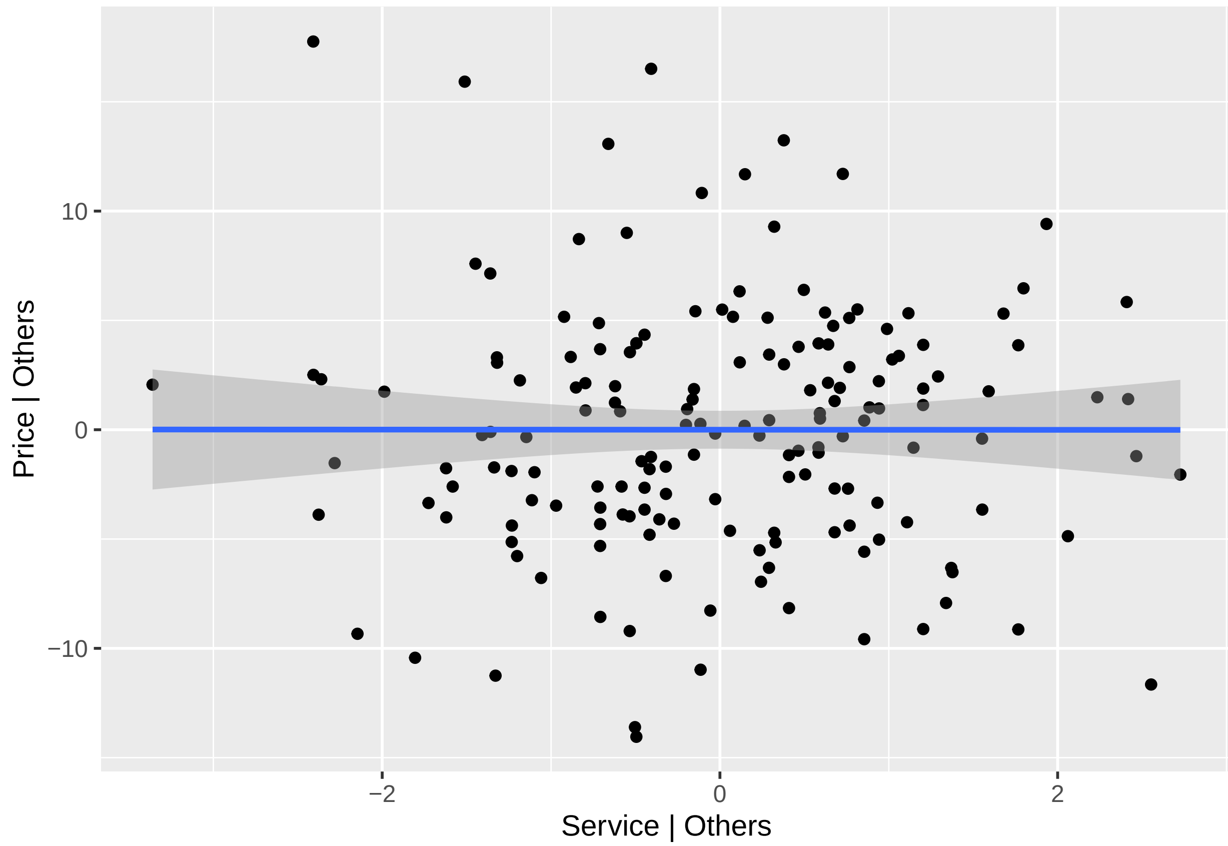
1. Build the model $Y \sim X_1 + \cdots + X_k$ and record these residuals as ϵ_y.
2. Build the model $Z \sim X_1 + \cdots + X_k$ and record these residuals as ϵ_z.
3. Fit the model $\epsilon_y \sim \epsilon_z$ and plot that model.

```r
m_y <- lm(Price ~ Food + Decor + East, data=nyc)     # Without Service
m_z <- lm(Service ~ Food + Decor + East, data=nyc)   # Service is the response!

avp.df <- data.frame( e_y = resid(m_y),
                      e_z = resid(m_z))

ggplot(avp.df, aes(y=e_y, x=e_z)) +
  geom_point() +
  geom_smooth(method='lm') +
  labs(y='Price | Others', x='Service | Others')
```

```
## `geom_smooth()` using formula 'y ~ x'
```



It is a little confusing why we should be interpreting the result of a regression of the residuals, but if we consider

- ϵ_y as the unaccounted for *variability* in the y after accounting for the $X_1, \dots, X_k$ variables
- ϵ_z as the remaining *signal* in z that hasn't already been accounted for by $X_1, \dots, X_k$

Then the regression of $\epsilon_y \sim \epsilon_z$ is exactly the correct model for interpreting the effect of Z after accounting for the effect of $X_1, \dots, X_k$.

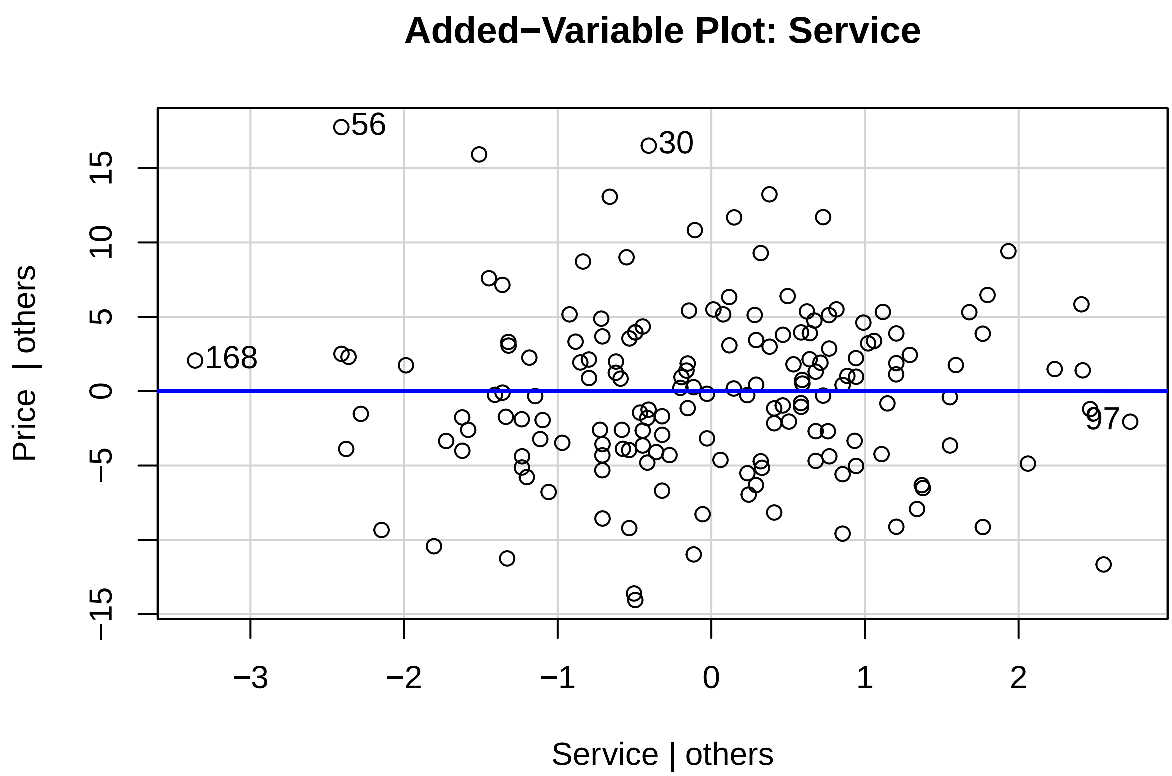
```
lm(e_y ~ e_z, data=avp.df) %>% summary()
```

```
##
## Call:
## lm(formula = e_y ~ e_z, data = avp.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.0465  -3.8837   0.0373   3.3942  17.7491
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.400e-16  4.387e-01   0.000    1.000
## e_z         -2.727e-03  3.926e-01  -0.007    0.994
##
## Residual standard error: 5.686 on 166 degrees of freedom
## Multiple R-squared:  2.907e-07,  Adjusted R-squared:  -0.006024
## F-statistic: 4.826e-05 on 1 and 166 DF,  p-value: 0.9945
```

Notice the `e_z` estimate, standard error, t-value, and p-value are all identical to the what we saw in the original coefficients table.

The creation of these graphs is a little annoying to do by hand and we could use the package `car` instead. This is what is most often done in "Learn to do statistics using R" style textbooks.

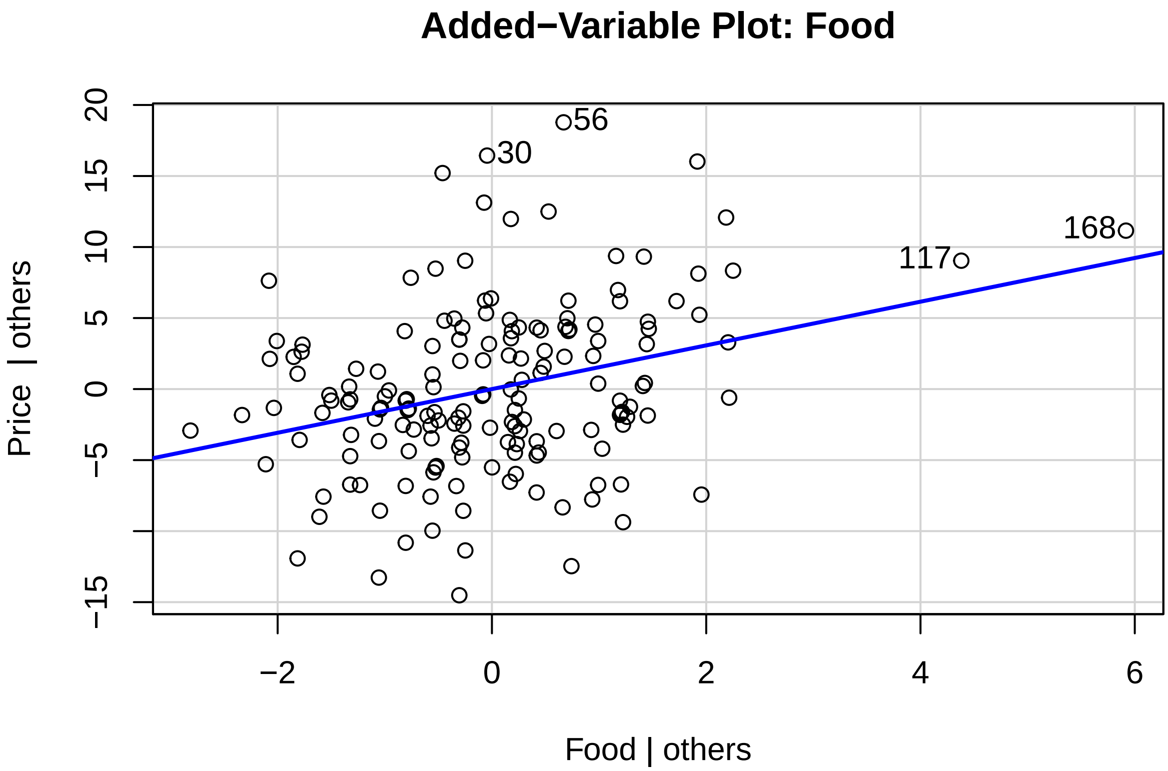
```
car::avPlot(m_full, 'Service')
```



What the x-axis represents is the *deviation* in the level of service you would expect to see after already accounting for a restaurants Price and Decor. So a negative value here doesn't mean that the service is bad, just less than you would have expected given the other covariates. Similarly the y-axis is the deviation from the expected price than you would have otherwise expected given the other covariates.

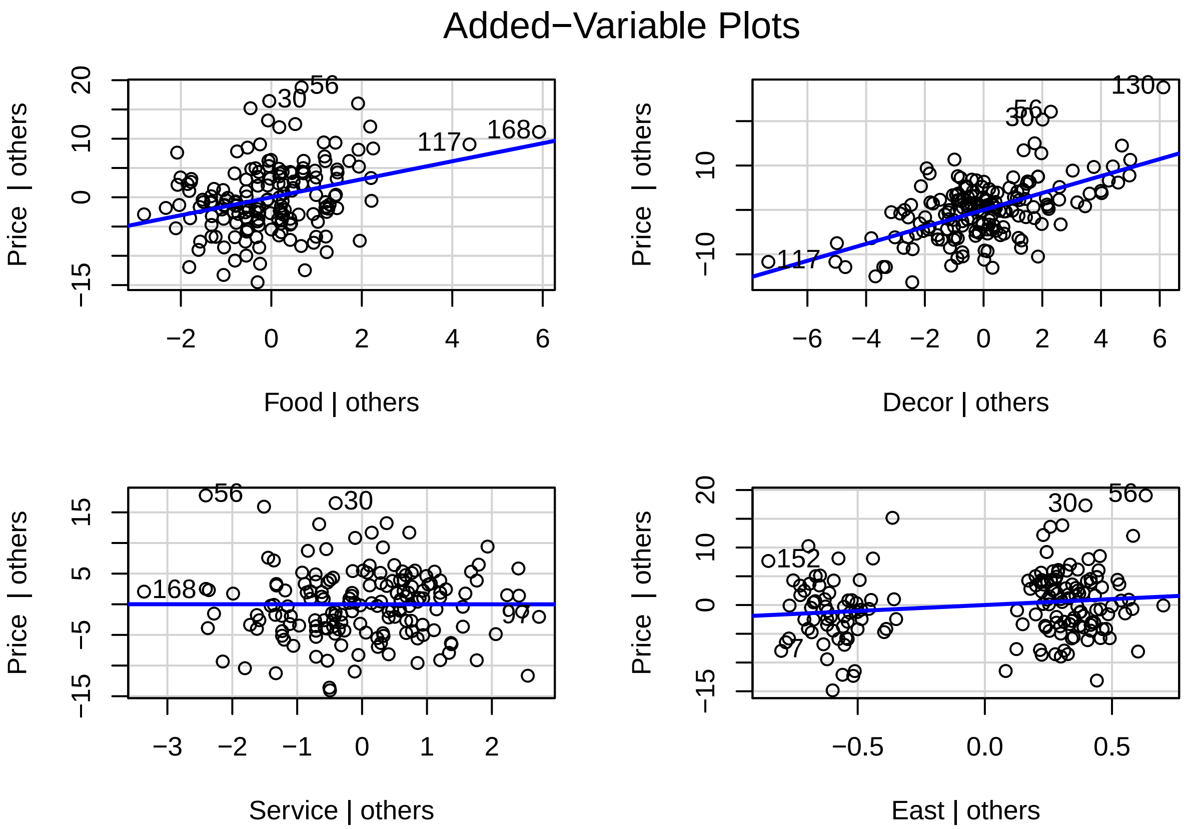
Notice that the plot for `Food` is surprising because there are two restaurants (rows 117 and 168) that have food quality WAY better than you would expect given the other variables. Furthermore rows 30 and 56 have prices MUCH higher than you would expect given the other variables and food quality.

```
car::avPlot(m_full, 'Food')   # show AVP for Service variable.
```



The added variable plot facilitates investigation of issues with the regression assumptions of linearity and homoskedasticity associated with a singular variate. These issues are more clearly visible when looking at the ADV than when looking at the pairs plots. To make it easy to graph all the added variable plots associated with a model we could use the car::avPlots() function.

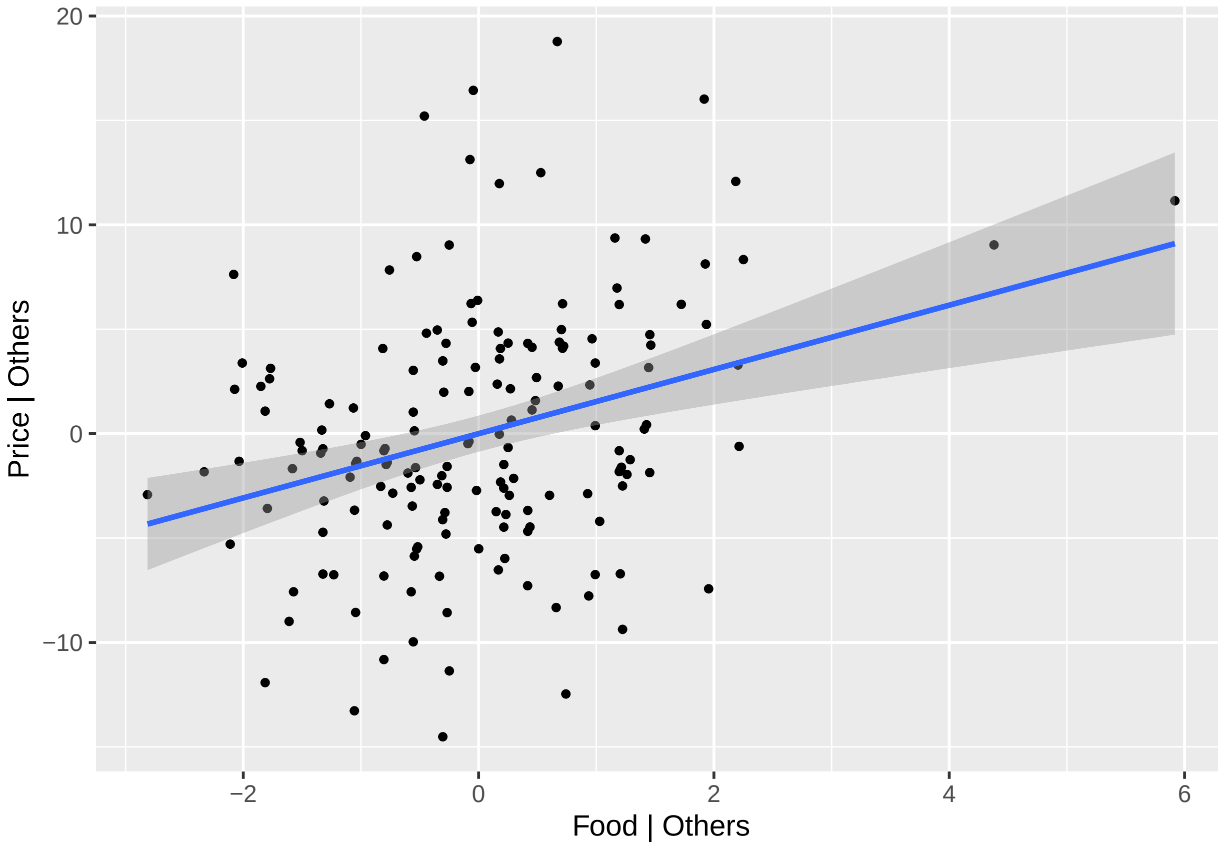
```
car::avPlots(m_full)
```



ggAVplots Package

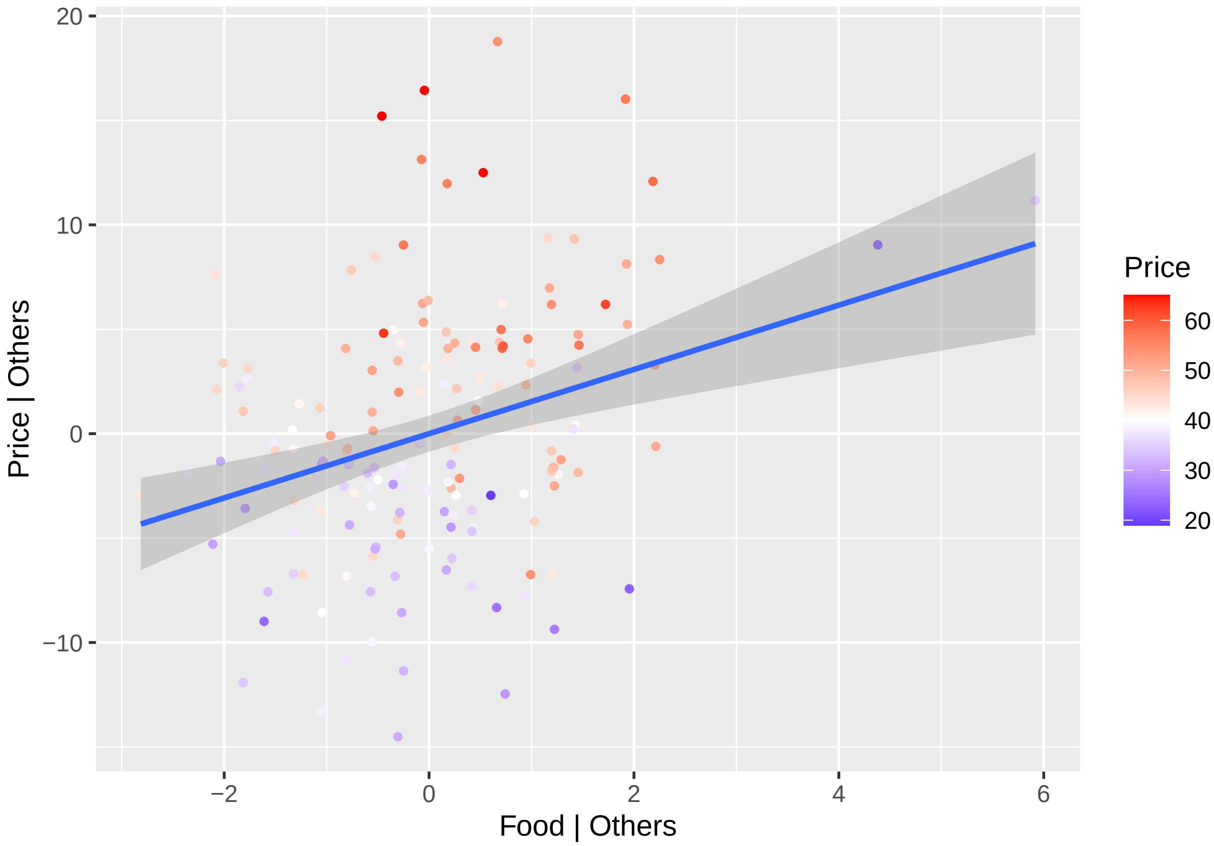
The `car::avPlot()` function is very convenient but it relies on base R graphics and also doesn't accommodated, for example, mixed-effects models. The package `ggAVplots` tries to account for that. Currently this package is available on GitHub.

```
#devtools::install_github('dereksonderegger/ggAVplots')
ggAVplots::ggAVplot(m_full, 'Food')   # identical to the car::avPlot()
```



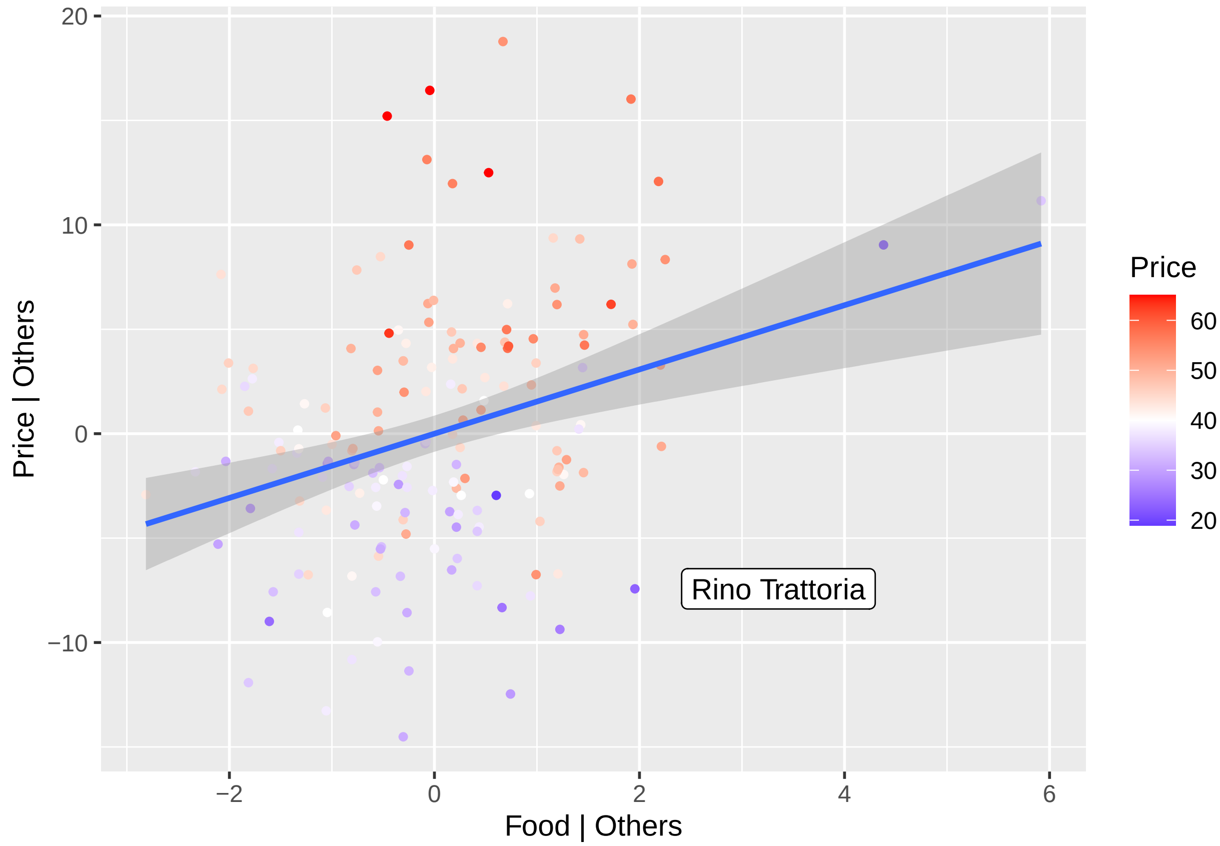
but it might be helpful to color code the points to include the raw prices. To do this, we have to include the data frame for the other covariates, and possibly covariates that are not included in the model (for example the restaurant names).

```
ggAVplots::ggAVplot(m_full, 'Food', data=nyc, color=Price) +
  scale_colour_gradient2(low='blue',mid='white',high='red', midpoint=40)
```



Notice there is a strong blue point near (e_z = 2, and e_y=-7). This restaurant has very low prices and better food than you expect given everything else. I'd love to label that restaurant...

```
ggAVplots::ggAVplot(m_full, 'Food', data=nyc, color=Price) +
  scale_colour_gradient2(low='blue',mid='white',high='red', midpoint=40) +
  geom_label(
    aes(label=Restaurant,
        x=e_z+1.4),                        # move label so not on the point
    color='black',                         # override the global Price coloring
    data= ~filter(., e_z>1.8, e_y < -7))   # just label points like this
```



A mixed effects model

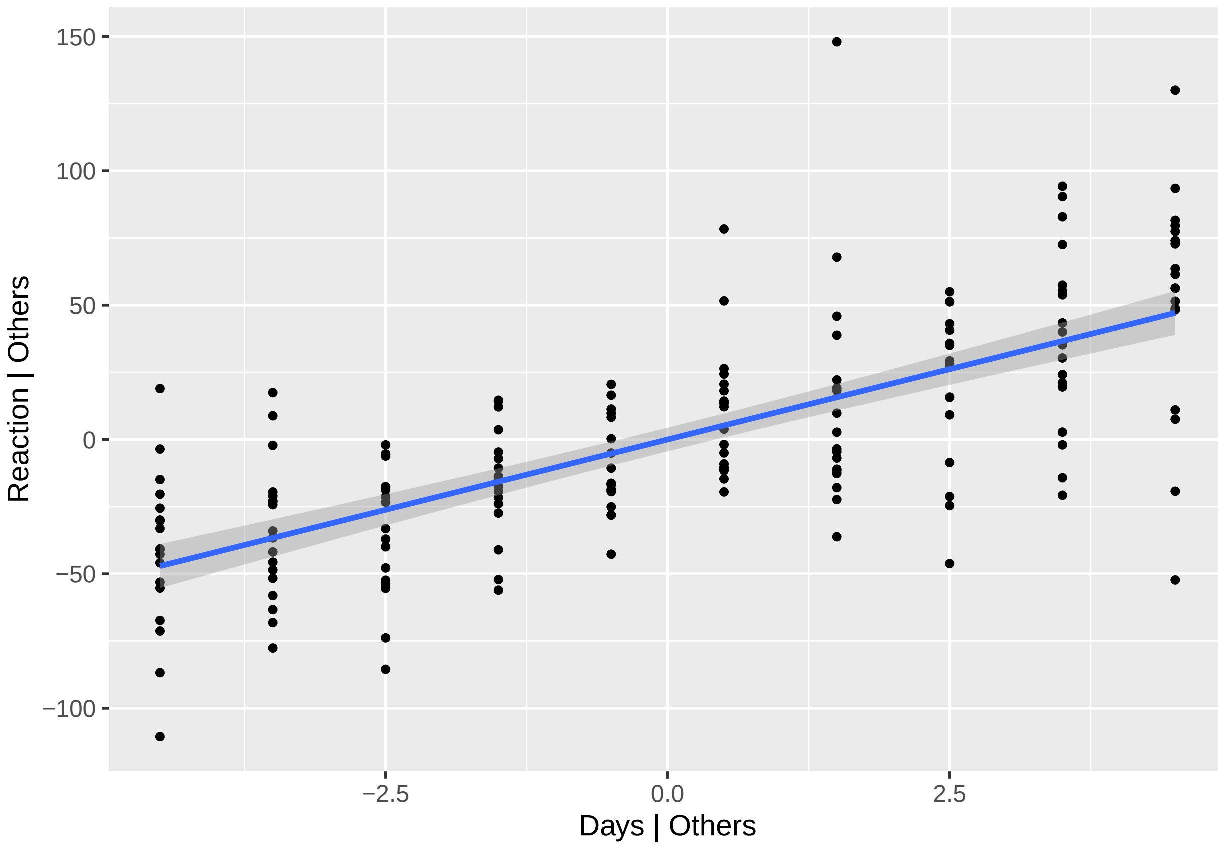
The `ggAVplots` package can deal with random effect models as well.

```
# A mixed-effects model
data('sleepstudy', package='lme4')
model <- lmerTest::lmer( Reaction ~ Days + (1|Subject), data=sleepstudy)
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car
```

```
# car::avPlot(model, 'Days') # Error, no applicable method for class lmerMod
ggAVplots::ggAVplot(model, 'Days')
```

```
## boundary (singular) fit: see ?isSingular
```


150
100
50
0
−50
−100
Reaction | Others
−2.5
0.0
2.5
Days | Others