hw_07_mlr

lisa liubovich

2024-03-19

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(broom)
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.3.2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

Conceptual Exercises

1.

received help from chatgpt: https://chat.openai.com/c/31b742a5-b995-4362-bc5f-259e08e8cfba

a.

Yes, it is possible to reformulate this model in terms of the general linear model by transforming X_{i2} using the logarithm.

Mathematically:

We can set:

$Y_i = \beta_0 + \beta_1\ X_{i1} + \beta_2 \log10\ X_{i2} + \beta_3\ X_{i1}^2 + \varepsilon_i$

to $Y_i = \beta_0 + \beta_1\ X_{i1} + \beta_2' X_{i2}' + \beta_3\ X_{i1}^2 + \varepsilon_i$

where $X_{i2}' = \log10 + X_{i2}$ and $\beta_2' = \beta_2$

b.

No, it is not possible to reformulate this model in terms of the general linear model. In this model, Y_i is defined as the error term times an exponential function of a linear combination of independent variables. This is not directly a linear model because of the exponential term. There is no simple transformation that will convert this into the form of the general linear model.

c.

Yes, it is possible to reformulate this model in terms of the general linear model by transforming Y_i .

Mathematically:

We can transform it as:

$Y_i = \log(\ \beta_1\ X_{i1}\ ) + \beta_2\ X_{i2} + \varepsilon_i$

to $Y_i = \log(\ \beta_1\ ) + \log(\ X_{i1}\ ) + \beta_2\ X_{i2} + \varepsilon_i$

to $Y_i = \beta_0 + \beta_1\text{'}\ \log(\ X_{i1}\ ) + \beta_2\ X_{i2} + \varepsilon_i$

where $\beta_1\text{'} = \log(\ \beta_1\ )$

2.

with help from chat gpt: https://chat.openai.com/c/31b742a5-b995-4362-bc5f-259e08e8cfba

```
# model <- lm(Y ~ X1 + I(2) + X3, data = df)
# model
```

Brand Preference

```
brand <- read_csv("https://dcgerard.github.io/stat_415_615/data/brand.csv")
```

```
## Rows: 16 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (3): like, moisture, sweetness
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

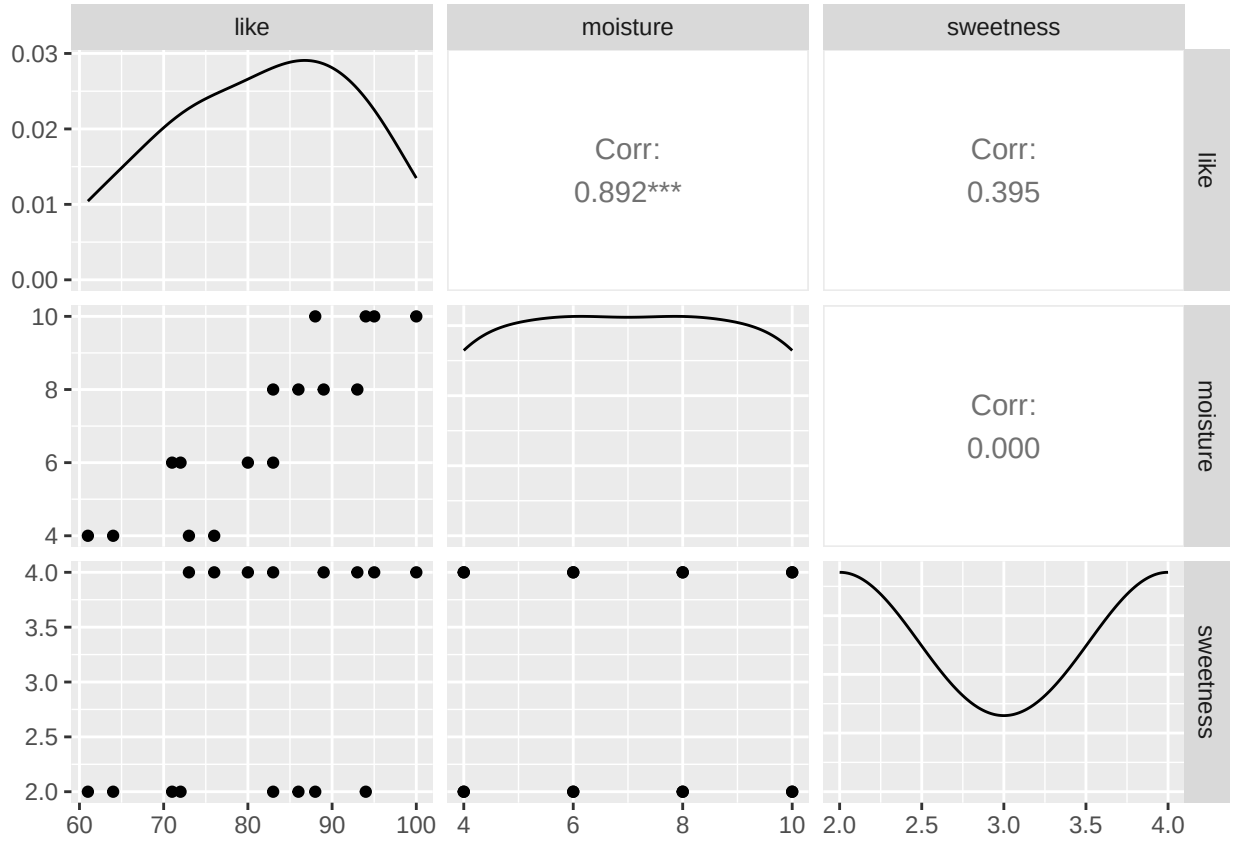
```
brand
```

```
## # A tibble: 16 x 3
##     like moisture sweetness
##    <dbl>    <dbl>     <dbl>
##  1    64        4         2
##  2    73        4         4
##  3    61        4         2
##  4    76        4         4
##  5    72        6         2
```

```
##  6    80         6          4
##  7    71         6          2
##  8    83         6          4
##  9    83         8          2
## 10    89         8          4
## 11    86         8          2
## 12    93         8          4
## 13    88        10          2
## 14    95        10          4
## 15    94        10          2
## 16   100        10          4
```

1.

```
ggpairs(brand)
```



This plot contains the visual representation of the relationship between like and moisture (which seems to be a positive linear relationship), like and sweetness (which appears to not be linear and have highly unequal variance), and moisture and sweetness (which also appears to not be linear and highly unequal variance).

2.

```r
lm_brand <- lm(like ~ moisture + sweetness, data = brand)
tidy(lm_brand, conf.int = TRUE)
```

```
## # A tibble: 3 x 7
##   term        estimate std.error statistic       p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>         <dbl>    <dbl>     <dbl>
## 1 (Intercept)    37.7      3.00      12.6  0.0000000120     31.2      44.1
## 2 moisture        4.42     0.301     14.7  0.00000000178     3.77      5.08
## 3 sweetness       4.37     0.673      6.50 0.0000201         2.92      5.83
```

Estimated regression function:

$Y_i = 37.650 + 4.425X_{i1} + 4.375X_{i2}$

where Y_i is the degree of brand liking for observations i, X_{i1} is the moisture content for observations i, and X_{i2} is the sweetness content for observations i.

3.

coefficient for moisture: 4.425 (95% CI: 3.774, 5.076)

Interpretation:

For every additional point of brand liking, the moisture content is 4.425 units higher on average (95% CI of 3.774 units higher to 5.076 units higher), while for sweetness.

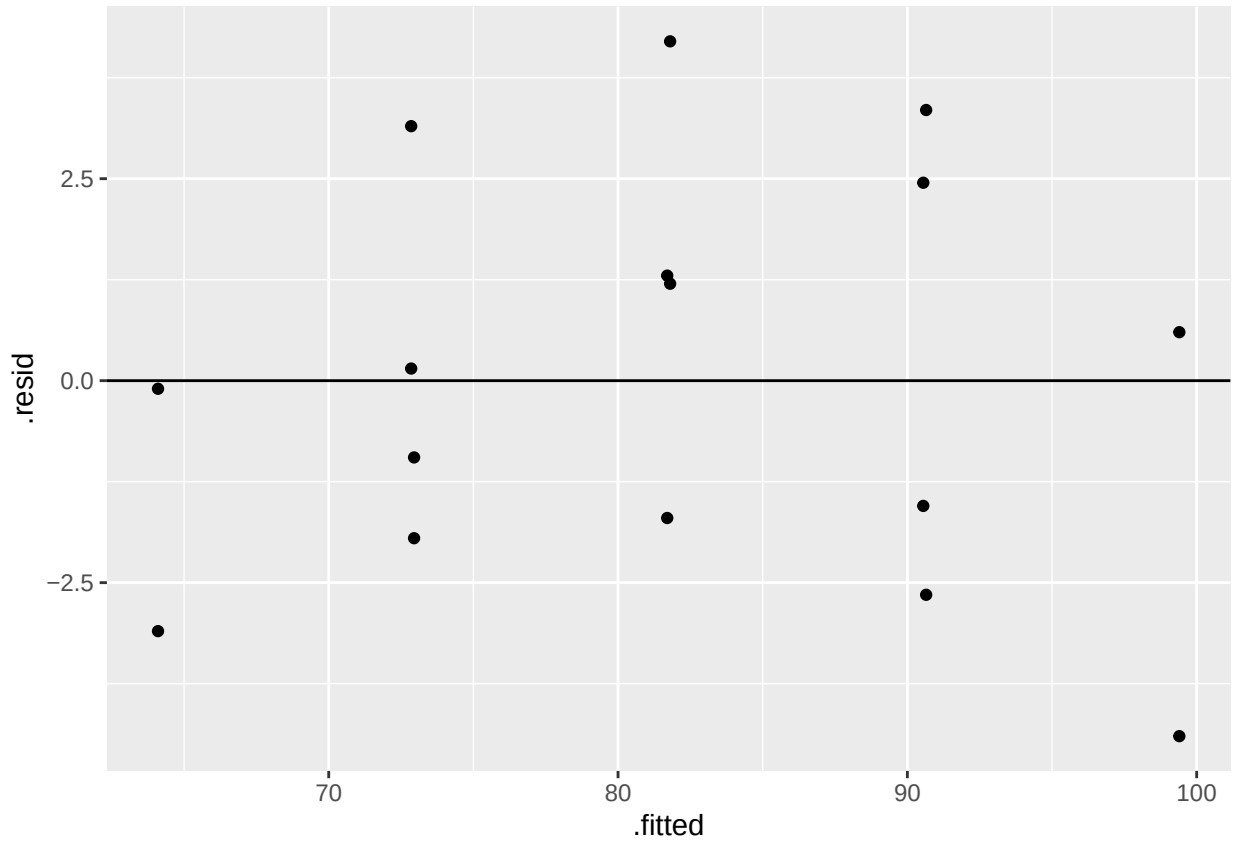
coefficient for sweetness: 4.375 (95% CI: 2.920, 5.830)

Interpretation:

For every additional point of brand liking, the sweetness content is 4.375 units higher on average (95% CI of 2.920 units higher to 5.830 units higher), adjusting for moisture.

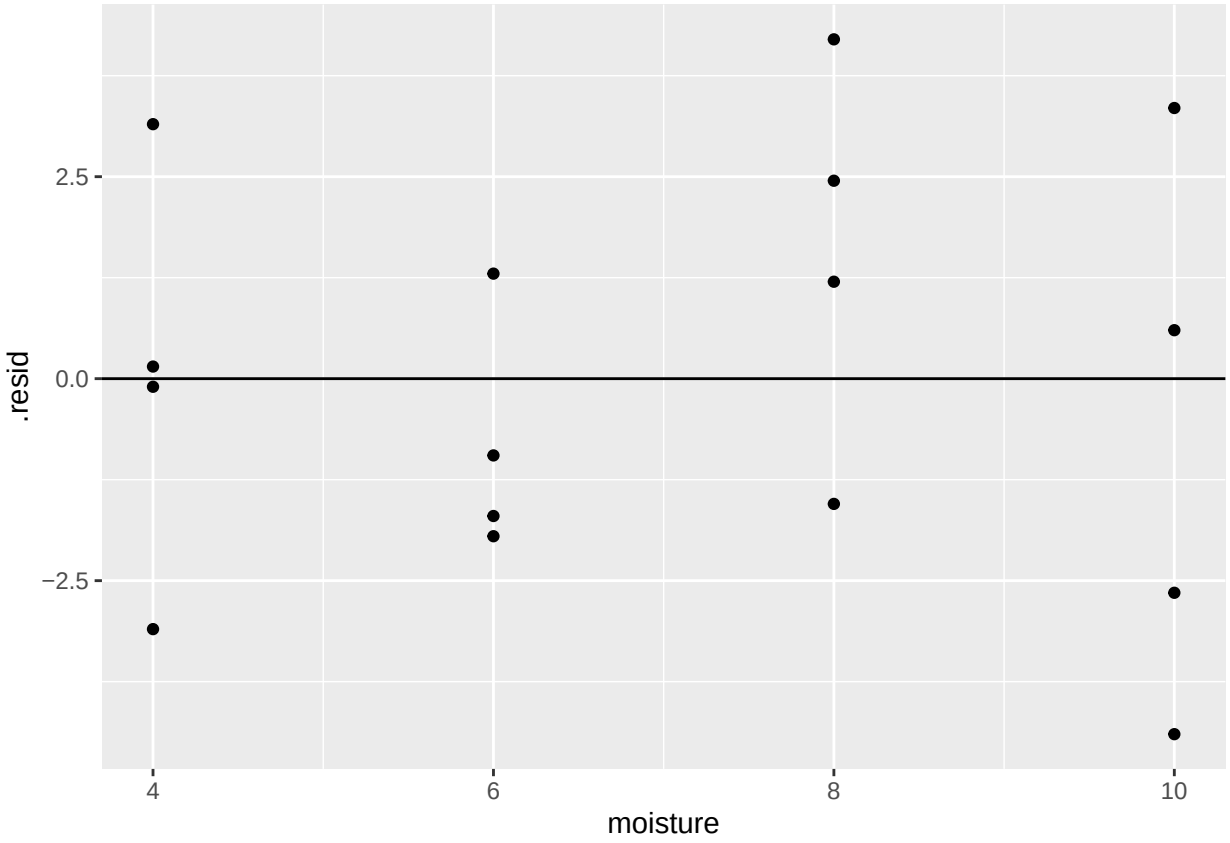
4.

```r
alm_brand <- augment(lm_brand)
ggplot(alm_brand, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0)
```



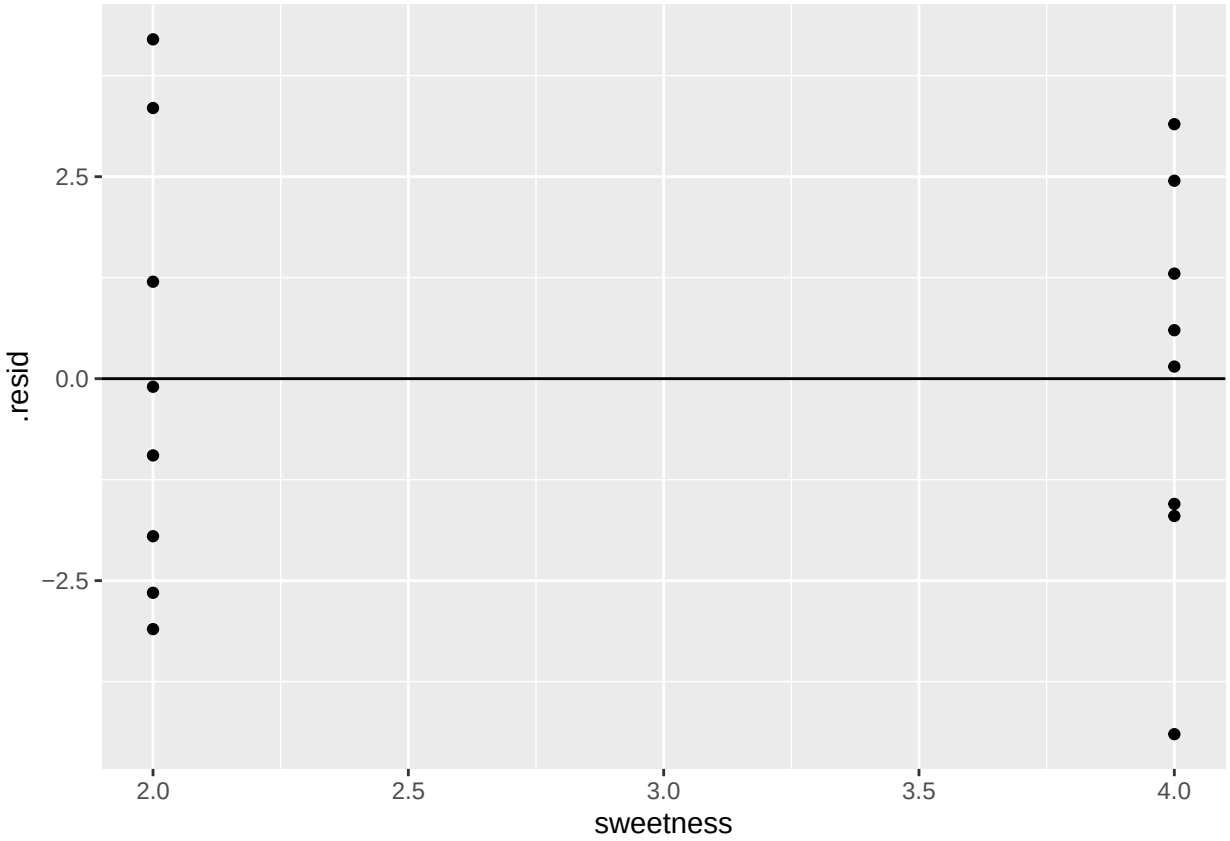
Seems to be curved and pretty unequal variance.

```
ggplot(alm_brand, aes(x = moisture, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0)
```



Super curved, super unequal variance.

```
ggplot(alm_brand, aes(x = sweetness, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0)
```



Relatively flat line, not super crazy differences in variance.

5.

```
newdf <- data.frame(sweetness = 3, moisture = 7)
predict(object = lm_brand, newdata = newdf, interval = "confidence")
```

```
##     fit      lwr      upr
## 1 81.75 80.29537 83.20463
```

Predicted liking score range: between 80.296 and 83.205 points.

Indicator Variables and Matrix Formulation

```
marry <- tribble(~happy, ~married,
73, "single",
79, "single",
72, "single",
58, "married",
72, "married",
74, "married",
```

```
77, "divorced",
51, "divorced",
63, "divorced")
marry
```

```
## # A tibble: 9 x 2
##   happy married
##   <dbl> <chr>
## 1    73 single
## 2    79 single
## 3    72 single
## 4    58 married
## 5    72 married
## 6    74 married
## 7    77 divorced
## 8    51 divorced
## 9    63 divorced
```

1.

Indicator variables:

X_{i1} is the predictor for being married (1 if married, 0 otherwise)

X_{i2} is the predictor for being divorced (1 if divorced, 0 otherwise)

that is,

married: $X_{i1} = 1$, $X_{i2} = 0$, divorced : $X_{i1} = 0$, $X_{i2} = 1$, single: $X_{i1} = 0$, $X_{i2} = 0$.

The linear model can be written as:

$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon_i$, where:

β_0 is the intercept term representing single individuals ($X_1 = 0$, $X_2 = 0$)

β_1 is the coefficient representing the difference in happiness between married and single individuals

β_2 is the coefficient representing the difference in happiness between divorced and single individuals

ε_i represents the error term with mean 0, constant variance, and uncorrelated.

2.

single person indicator: $X_1 = 0$, $X_2 = 0$

the model is:

$Y_i = \beta_0 + \beta_1 (0) + \beta_2 (0) + \varepsilon_i$ –> $Y_i = \beta_0 + \varepsilon_i$

where β_0 is the average level of happiness for single people.

3.

with help from chat gpt for LaTeX: https://chat.openai.com/c/31b742a5-b995-4362-bc5f-259e08e8cfba

$$X = \begin{bmatrix} 1 & X_{11} & X_{12} \\ 1 & X_{21} & X_{22} \\ 1 & X_{31} & X_{32} \\ 1 & X_{41} & X_{42} \\ 1 & X_{51} & X_{52} \\ 1 & X_{61} & X_{62} \\ 1 & X_{71} & X_{72} \\ 1 & X_{81} & X_{82} \\ 1 & X_{91} & X_{92} \end{bmatrix}$$

4.

with help from chatgpt for syntax: https://chat.openai.com/c/31b742a5-b995-4362-bc5f-259e08e8cfba

```
marry1 <- marry %>%
  mutate(married_ind = ifelse(married == "married", 1, 0),
         divorced_ind = ifelse(married == "divorced", 1, 0))
lm_marry <- lm(happy ~ married_ind + divorced_ind, data = marry1)
tidy(lm_marry, conf.int = TRUE)
```

```
## # A tibble: 3 x 7
##   term         estimate std.error statistic    p.value conf.low conf.high
##   <chr>           <dbl>     <dbl>     <dbl>      <dbl>    <dbl>     <dbl>
## 1 (Intercept)     74.7       5.37    13.9   0.00000863     61.5     87.8
## 2 married_ind     -6.67      7.60    -0.878 0.414         -25.3     11.9
## 3 divorced_ind   -11         7.60    -1.45  0.198         -29.6      7.59
```

Model for divorced: $Y_i = \beta_0 + \beta_1 \ (\ X_{i1} = 0) + \beta_2 \ (\ X_{i2} = 1 \ ) + \varepsilon_i$ –> $Y_i = \beta_0 + \beta_2 + \varepsilon_i$, where β_2 is the average difference in happiness between divorced and single individuals.

On average, divorced people have a happiness level that is 11 points lower than single individuals.