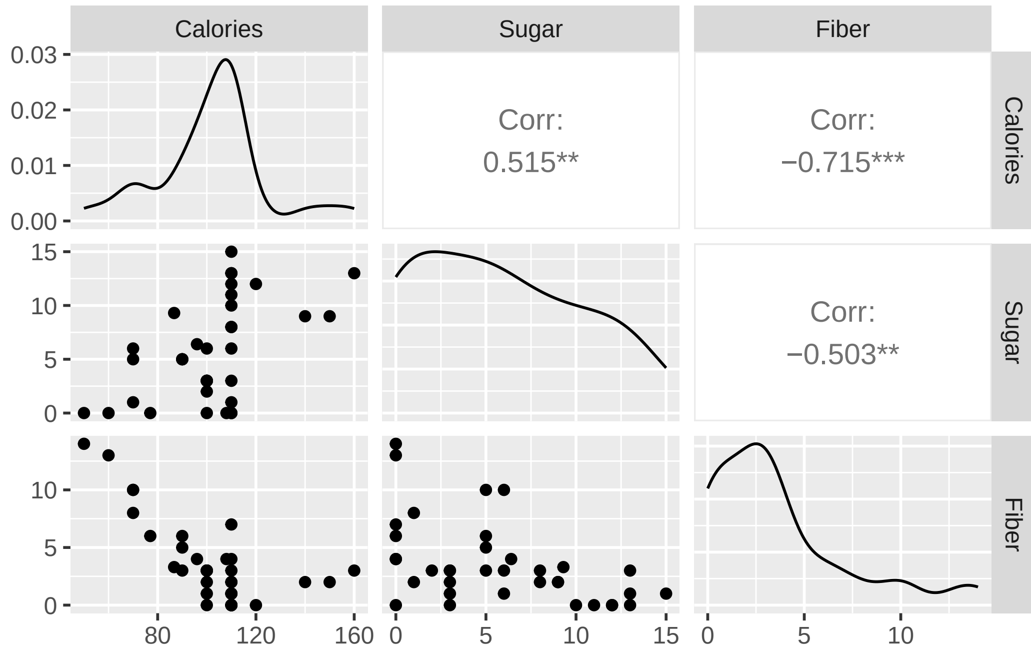
Homework 4

Explore 5 Data Sets to Help with classes

2 From Statcalpolypackage 2 from base R 1 from another place

```
library(statcalpolypackage)                  ## Data Extraction
library(gato365dsh2024)                      ## Data Extraction
library(dplyr)                               ## Data Transformation
library(ggplot2)                             ## Data Visualization
library(GGally)                              ## Data Visualization
library(broom)                               ## Data Analysis
```

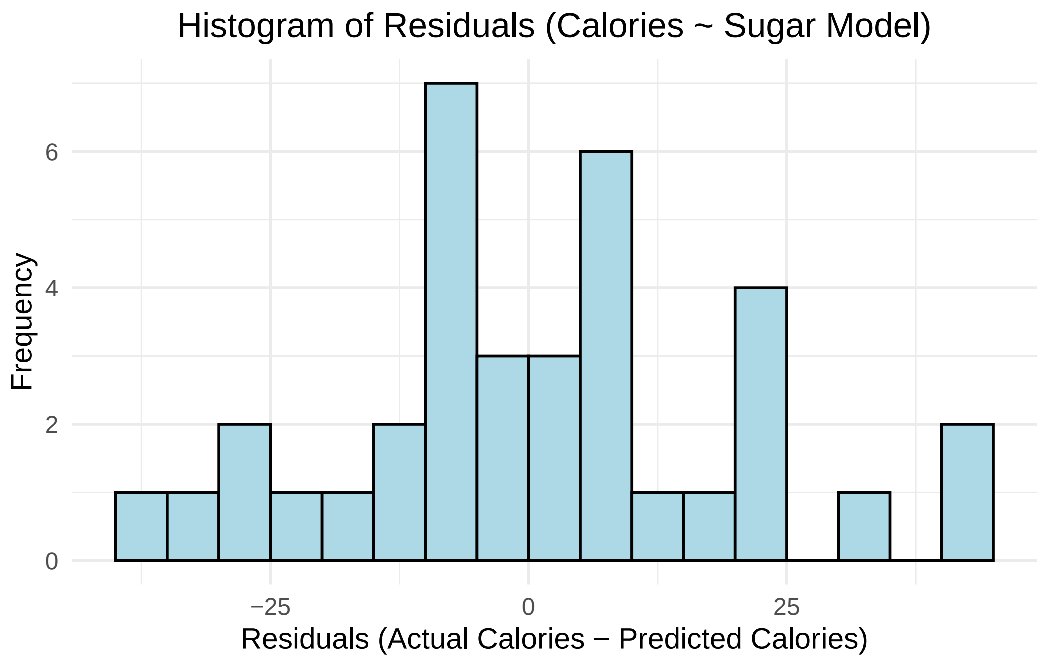
```
Cereal %>%
  select(-Cereal) %>%
  ggpairs()
```




```r
# Fit the linear model using the lm() function
lm_cereal <- lm(Calories ~ Sugar, data = Cereal)
summary(lm_cereal)
```

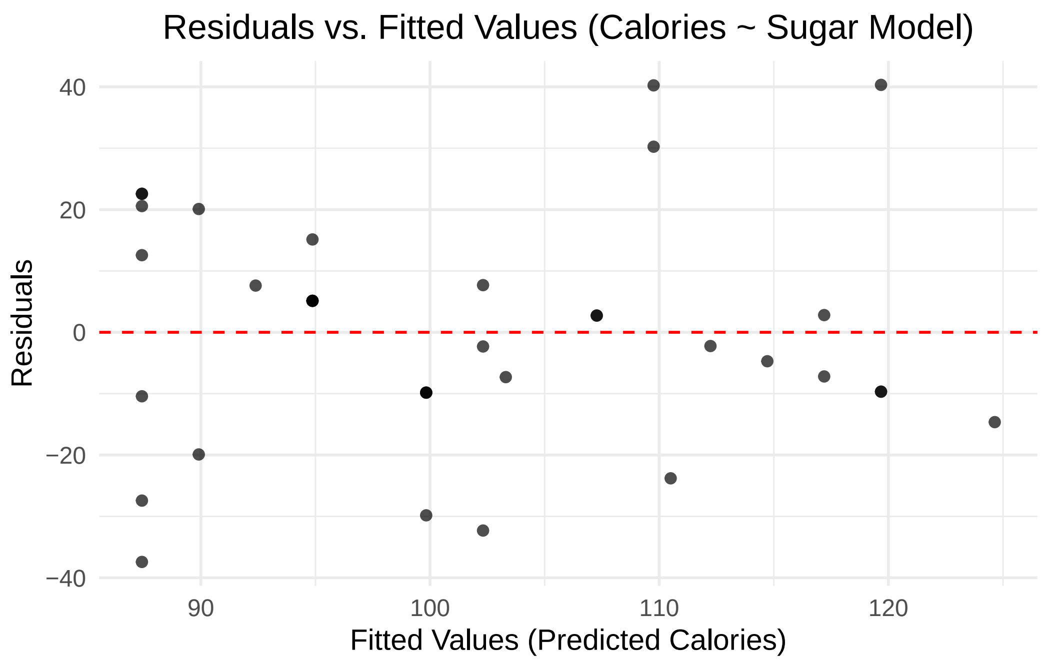
```
Call:
lm(formula = Calories ~ Sugar, data = Cereal)

Residuals:
    Min      1Q  Median      3Q     Max
-37.428  -9.832   0.245   8.909  40.322

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  87.4277     5.1627  16.935   <2e-16 ***
Sugar         2.4808     0.7074   3.507   0.0013 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 19.27 on 34 degrees of freedom
Multiple R-squared:  0.2656,	Adjusted R-squared:  0.244
F-statistic:  12.3 on 1 and 34 DF,  p-value: 0.001296
```

```r
augmented_cereal <- augment(lm_cereal)

# Plot histogram of residuals
# We use the.resid column from the augmented data frame.
ggplot(augmented_cereal, aes(x =.resid)) +
  geom_histogram(binwidth = 5, fill = "lightblue", color = "black", boundary = 0) + # Adju
  labs(title = "Histogram of Residuals (Calories ~ Sugar Model)",
       x = "Residuals (Actual Calories - Predicted Calories)",
       y = "Frequency") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




```r
# Scatter plot of Residuals vs. Fitted values
# This plot helps check for non-constant variance (heteroscedasticity - look for funnel sh
# and remaining non-linearity (look for curved patterns).
ggplot(augmented_cereal, aes(x =.fitted, y =.resid)) +
  geom_point(alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") + # Reference line at zer
  labs(title = "Residuals vs. Fitted Values (Calories ~ Sugar Model)",
       x = "Fitted Values (Predicted Calories)",
       y = "Residuals") +
```

```r
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



(Answer Key)

```r
n_cereal <- nrow(Cereal)
p_cereal <- 1 # Intercept + Sugar coefficient
leverage_threshold_cereal <- 2 * p_cereal / n_cereal
cooks_threshold_cereal <- 4 / n_cereal
resid_sd_cereal <- sd(augmented_cereal$.resid)
```

Leverage (Answer Key)

```r
leverage_threshold_cereal
```

```
[1] 0.05555556
```

Cooks Distance (Answer Key)

```r
cooks_threshold_cereal
```

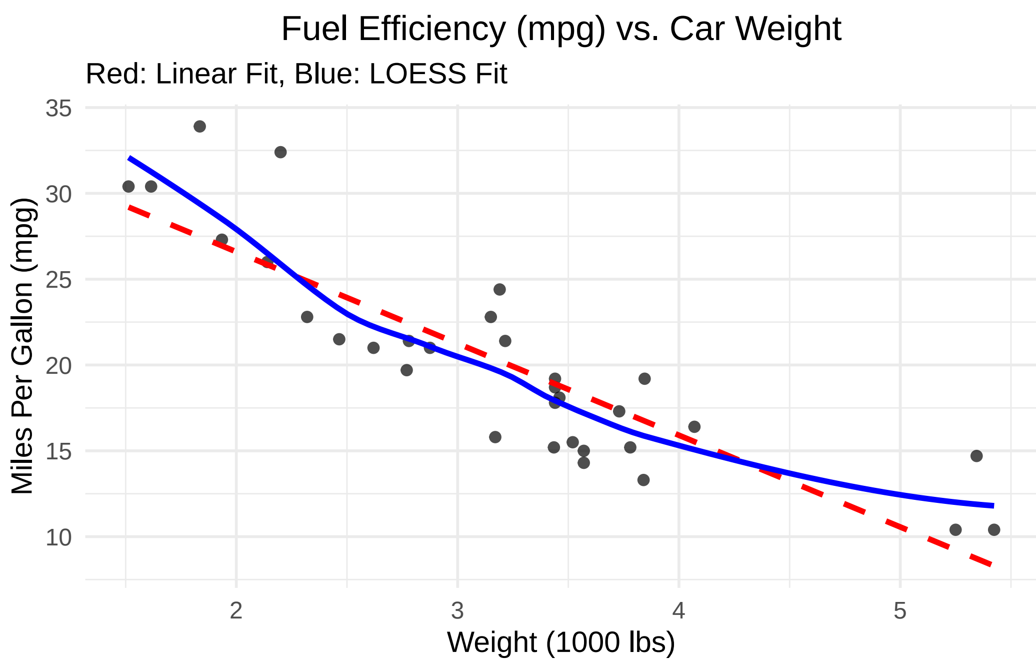
```
[1] 0.1111111
```

```r
augmented_cereal %>%
  arrange(desc(.cooksd)) %>%
  head(10) %>%
  round(3)
```

```
# A tibble: 10 x 8
   Calories Sugar .fitted .resid  .hat .sigma .cooksd .std.resid
      <dbl> <dbl>   <dbl>  <dbl> <dbl>  <dbl>   <dbl>      <dbl>
 1      160    13    120.   40.3 0.099   18.1   0.268      2.20
 2       50     0    87.4  -37.4 0.072   18.4   0.157     -2.02
 3      150     9    110.   40.2 0.042   18.2   0.101      2.13
 4       60     0    87.4  -27.4 0.072   18.9   0.084     -1.48
 5      110     0    87.4   22.6 0.072   19.1   0.057      1.22
 6      110     0    87.4   22.6 0.072   19.1   0.057      1.22
 7      140     9    110.   30.2 0.042   18.8   0.057      1.60
 8      110    15    125.  -14.6 0.144   19.4   0.057     -0.821
 9      108     0    87.4   20.6 0.072   19.2   0.047      1.11
10       70     6    102.  -32.3 0.028   18.7   0.041     -1.70
```

```r
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed") + # Add linea
  geom_smooth(method = "loess", se = FALSE, color = "blue") + # Add non-linear trend (LOES
  labs(title = "Fuel Efficiency (mpg) vs. Car Weight",
       subtitle = "Red: Linear Fit, Blue: LOESS Fit",
       x = "Weight (1000 lbs)",
       y = "Miles Per Gallon (mpg)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

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



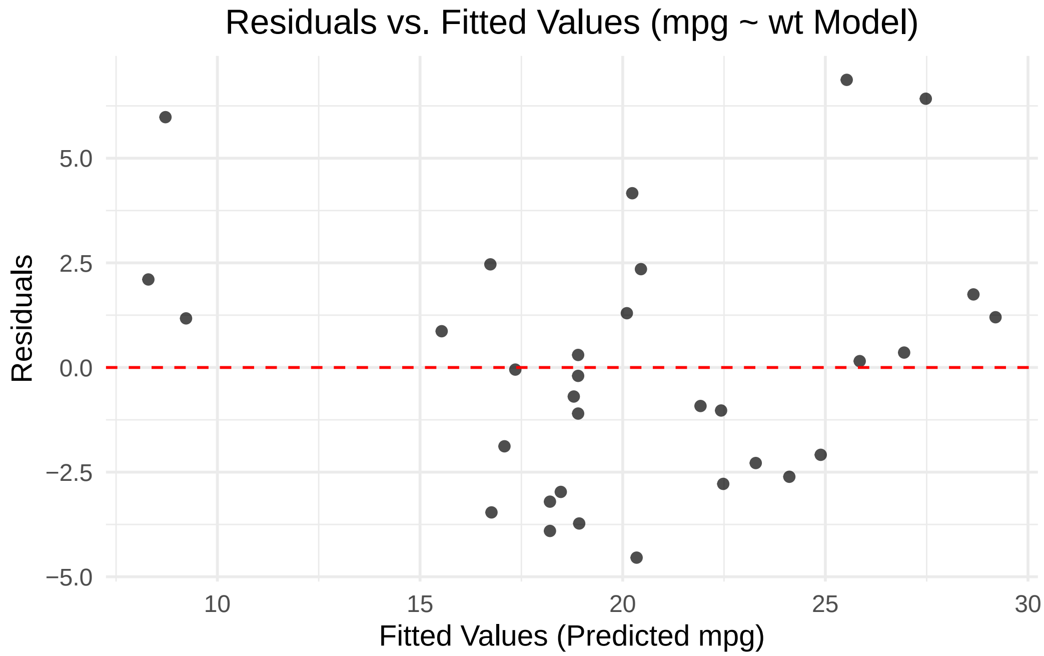
Explain Lowess

```
# Fit the initial linear model
lm_mtcars_orig <- lm(mpg ~ wt, data = mtcars_data)

# Display the model summary
summary(lm_mtcars_orig)
```

```
Call:
lm(formula = mpg ~ wt, data = mtcars_data)

Residuals:
    Min      1Q  Median      3Q     Max
-4.5432 -2.3647 -0.1252  1.4096  6.8727

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
wt           -5.3445     0.5591  -9.559 1.29e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared:  0.7528,	Adjusted R-squared:  0.7446
F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

```
augmented_mtcars <- augment(lm_mtcars_orig)
# Scatter plot of Residuals vs. Fitted values
# This plot helps check for non-constant variance (heteroscedasticity - look for funnel sh
# and remaining non-linearity (look for curved patterns).
ggplot(augmented_mtcars, aes(x =.fitted, y =.resid)) +
  geom_point(alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") + # Reference line at zer
  labs(title = "Residuals vs. Fitted Values (mpg ~ wt Model)",
       x = "Fitted Values (Predicted mpg)",
       y = "Residuals") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



(Answer Key)

```
n_mtcars <- nrow(mtcars)
p_mtcars <- 1 # Intercept + Sugar coefficient
leverage_threshold_mtcars <- 2 * p_mtcars / n_mtcars
cooks_threshold_mtcars <- 4 / n_mtcars
```

Leverage (Answer Key)

```
leverage_threshold_mtcars
```

```
[1] 0.0625
```

Cooks Distance (Answer Key)

```
cooks_threshold_mtcars
```

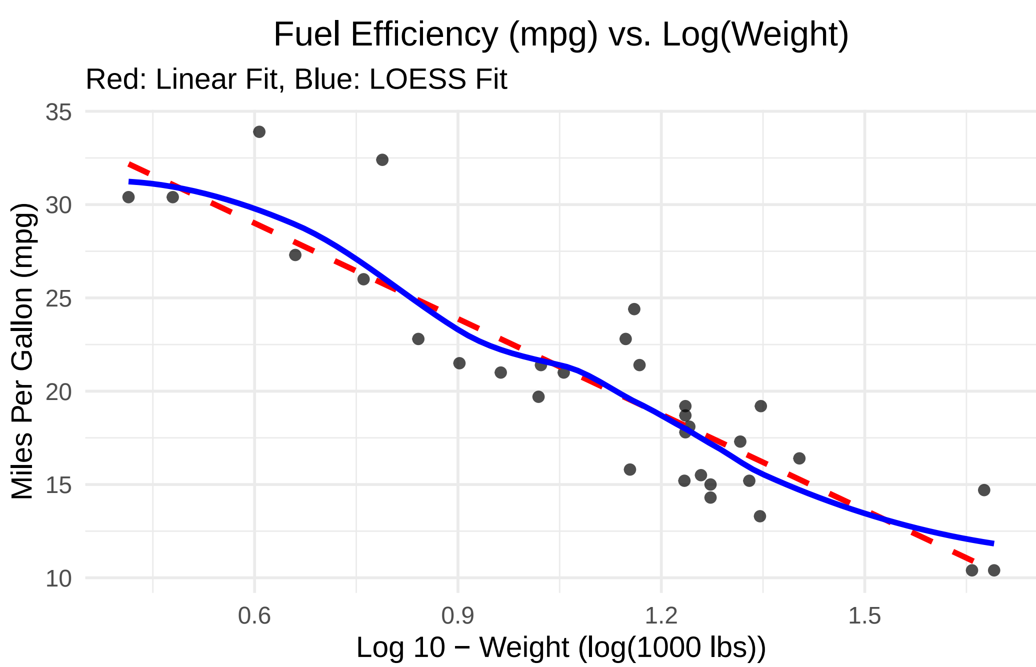
```
[1] 0.125
```

```
augmented_mtcars %>%
  arrange(desc(.cooksd)) %>%
  head(10) %>%
  round(3)
```

```
# A tibble: 10 x 8
     mpg    wt .fitted .resid   .hat .sigma .cooksd .std.resid
   <dbl> <dbl>   <dbl>  <dbl>  <dbl>  <dbl>   <dbl>      <dbl>
 1  14.7  5.34    8.72   5.98 0.184   2.84   0.532       2.17
 2  33.9  1.84   27.5    6.42 0.096   2.83   0.26        2.22
 3  32.4  2.2    25.5    6.87 0.066   2.80   0.193       2.34
 4  10.4  5.42    8.30   2.10 0.195   3.07   0.072       0.77
 5  15.8  3.17   20.3   -4.54 0.031   2.98   0.037      -1.52
 6  13.3  3.84   16.8   -3.46 0.044   3.03   0.031      -1.16
 7  14.3  3.57   18.2   -3.90 0.035   3.01   0.031      -1.31
 8  24.4  3.19   20.2    4.16 0.031   3.00   0.031       1.39
 9  15.2  3.44   18.9   -3.73 0.033   3.02   0.026      -1.24
10  30.4  1.62   28.7    1.75 0.118   3.08   0.025       0.61
```

```r
# Create a new variable log_wt using dplyr::mutate
mtcars_data <- mtcars %>%
  mutate(log_wt = log(wt))

# Visualize the relationship with the transformed variable: mpg vs. log(wt)
# Again, show both linear and LOESS fits to assess if linearity improved.
ggplot(mtcars_data, aes(x = log_wt, y = mpg)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed") +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +
  labs(title = "Fuel Efficiency (mpg) vs. Log(Weight)",
       subtitle = "Red: Linear Fit, Blue: LOESS Fit",
       x = "Log 10 - Weight (log(1000 lbs))",
       y = "Miles Per Gallon (mpg)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

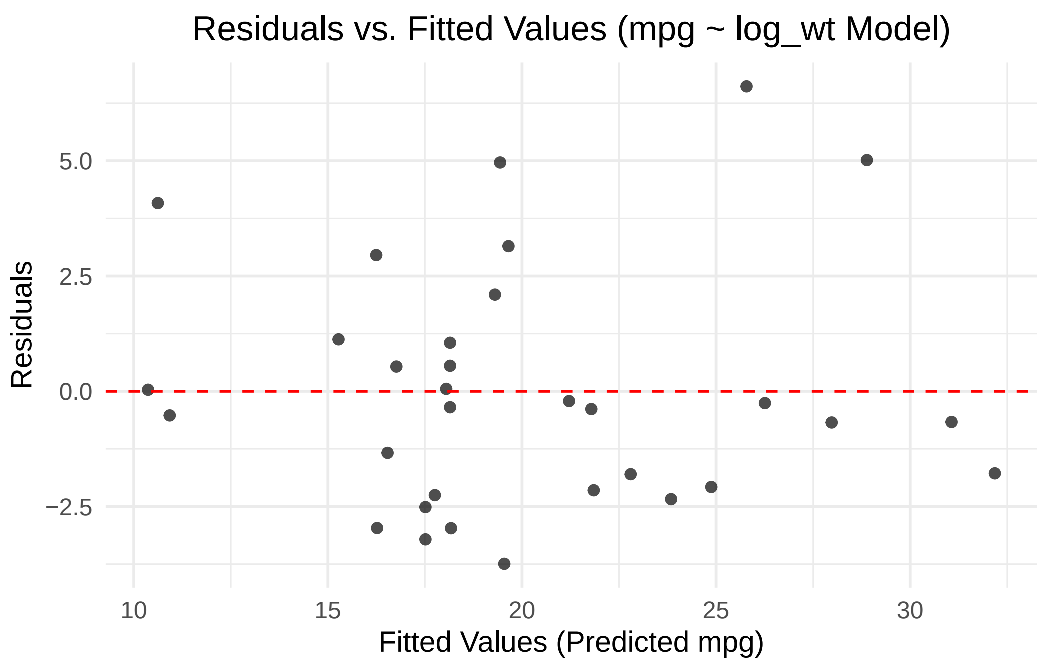
```
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
```



```r
# Fit the linear model using log_wt as the predictor
lm_mtcars_transformed <- lm(mpg ~ log_wt, data = mtcars_data)
# Display the model summary
summary(lm_mtcars_transformed)
```

```
Call:
lm(formula = mpg ~ log_wt, data = mtcars_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.7440 -2.0954 -0.3672  1.0709  6.6150 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   39.257      1.758   22.32  < 2e-16 ***
log_wt       -17.086      1.510  -11.31 2.39e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.669 on 30 degrees of freedom
Multiple R-squared:  0.8101,	Adjusted R-squared:  0.8038 
F-statistic:   128 on 1 and 30 DF,  p-value: 2.391e-12
```

```r
augmented_mtcars_transformed <- augment(lm_mtcars_transformed)
# Scatter plot of Residuals vs. Fitted values
# This plot helps check for non-constant variance (heteroscedasticity - look for funnel sh
# and remaining non-linearity (look for curved patterns).
ggplot(augmented_mtcars_transformed, aes(x =.fitted, y =.resid)) +
  geom_point(alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") + # Reference line at zer
  labs(title = "Residuals vs. Fitted Values (mpg ~ log_wt Model)",
       x = "Fitted Values (Predicted mpg)",
       y = "Residuals") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



(Answer Key)

```r
n_mtcars <- nrow(mtcars)
p_mtcars <- 1 # Intercept + Sugar coefficient
leverage_threshold_mtcars <- 2 * p_mtcars / n_mtcars
cooks_threshold_mtcars <- 4 / n_mtcars
```

Leverage (Answer Key)

```r
leverage_threshold_mtcars
```

```
[1] 0.0625
```

Cooks Distance (Answer Key)

```r
cooks_threshold_mtcars
```

```
[1] 0.125
```

```r
augmented_mtcars_transformed %>%
  select(-.rownames) %>%
  arrange(desc(.cooksd)) %>%
  head(10) %>%
  round(3)
```

```
# A tibble: 10 x 8
     mpg log_wt .fitted .resid  .hat .sigma .cooksd .std.resid
   <dbl>  <dbl>   <dbl>  <dbl> <dbl>  <dbl>   <dbl>      <dbl>
 1  33.9  0.607    28.9   5.01 0.116   2.53   0.262      2.00
 2  32.4  0.788    25.8   6.62 0.067   2.40   0.235      2.56
 3  14.7  1.68     10.6   4.08 0.13    2.59   0.2        1.64
 4  30.4  0.414    32.2  -1.78 0.191   2.69   0.065     -0.742
 5  24.4  1.16     19.4   4.96 0.032   2.55   0.058      1.89
 6  15.8  1.15     19.5  -3.74 0.032   2.62   0.033     -1.42
 7  13.3  1.34     16.3  -2.97 0.047   2.66   0.032     -1.14
 8  19.2  1.35     16.2   2.95 0.047   2.66   0.032      1.13
 9  14.3  1.27     17.5  -3.21 0.039   2.65   0.03      -1.23
10  15.2  1.23     18.2  -2.97 0.035   2.66   0.023     -1.13
```

How many points were above the line