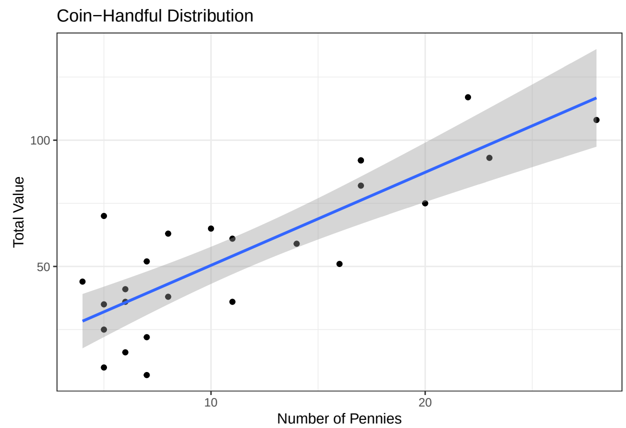
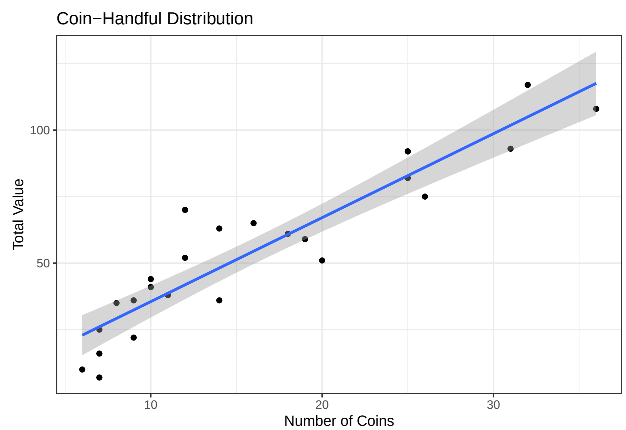
Interpretation of Multiple Regression Coefficients: Output

Brooke Coneeny

The data and code for this lecture can be found at my github: @brooke-coneeny






```
#Fitting a simple regression of value in cents on number of coins
model1 <- lm(value.in.cents ~ X..of.coins, data = coin_data)
#Saving the residuals of the above model
model1_resids <- resid(model1)
#Adding these residuals to the data set
coin_data <- coin_data %>% cbind(model1_resids)

#Fitting a simple regression of number of pennies on number of coins
model2 <- lm(X..of.pennies ~ X..of.coins, data = coin_data)
#Saving the residuals of the above model
model2_resids <- resid(model2)
#Adding these residuals to the data set
```

```r
coin_data <- coin_data %>% cbind(model2_resids)

#Fitting a simple regression of the first set of residuals on the second set of residuals
model3 <- lm(model1_resids ~ model2_resids, data = coin_data)
#This is the multiple regression
model4 <- lm(value.in.cents ~ X..of.pennies + X..of.coins, data = coin_data)
#This is the simple regression of x1 on y
model5 <- lm(value.in.cents ~ X..of.pennies, data = coin_data)

summary(model3)
```

```
##
## Call:
## lm(formula = model1_resids ~ model2_resids, data = coin_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.191  -2.679   1.098   3.336   6.057
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.109e-15  9.403e-01    0.00        1
## model2_resids -7.028e+00  6.428e-01  -10.93 1.38e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.702 on 23 degrees of freedom
## Multiple R-squared:  0.8386, Adjusted R-squared:  0.8316
## F-statistic: 119.5 on 1 and 23 DF,  p-value: 1.385e-10
```

```r
summary(model4)
```

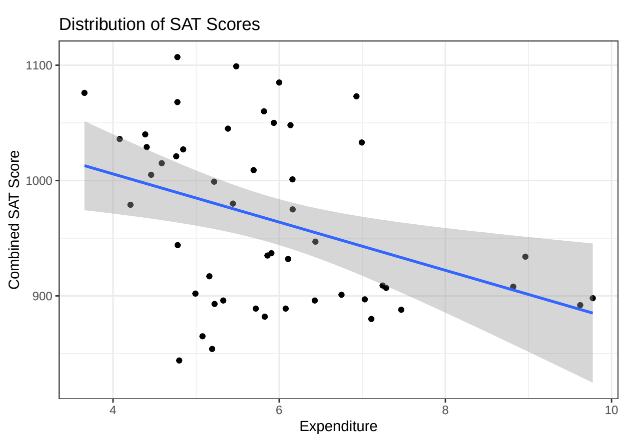
```
##
## Call:
## lm(formula = value.in.cents ~ X..of.pennies + X..of.coins, data = coin_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.191  -2.679   1.098   3.336   6.057
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.7493     2.1167  -0.826    0.417
## X..of.pennies  -7.0278     0.6573 -10.693 3.52e-10 ***
## X..of.coins     8.4026     0.5030  16.705 5.53e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.807 on 22 degrees of freedom
## Multiple R-squared:  0.9769, Adjusted R-squared:  0.9748
## F-statistic: 464.7 on 2 and 22 DF,  p-value: < 2.2e-16
```

```r
summary(model5)
```

```
##
## Call:
## lm(formula = value.in.cents ~ X..of.pennies, data = coin_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.394 -12.276   0.289  14.556  37.973
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    13.6114     6.8976   1.973   0.0606 .
## X..of.pennies   3.6832     0.5225   7.049  3.5e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.39 on 23 degrees of freedom
## Multiple R-squared:  0.6836, Adjusted R-squared:  0.6699
## F-statistic: 49.69 on 1 and 23 DF,  p-value: 3.498e-07
```

```r
initial_plot <- SAT_data %>%
  ggplot(aes(x = expenditure, y = Combined_SAT)) +
  geom_point() +
  geom_smooth(method = 'lm', formula = y ~ x) +
  theme_bw() +
  labs(
    title = "Distribution of SAT Scores",
    x = "Expenditure",
    y = "Combined SAT Score"
  )

initial_plot
```



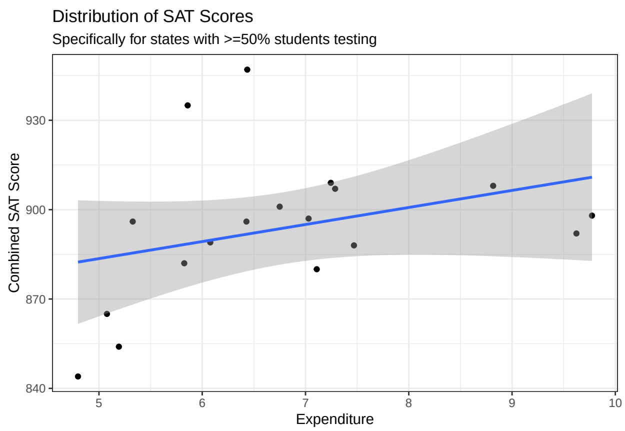

```r
grouped_SAT_data <- SAT_data %>%
  mutate(division = case_when(
    Pct_Taking_SAT >= 50 ~ "upper",
```

```
    Pct_Taking_SAT < 50 ~ "lower"
  ))
```

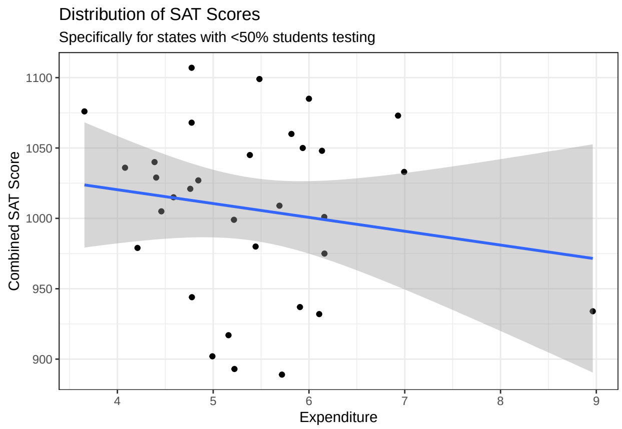
```
upper_plot <- grouped_SAT_data %>%
  filter(division == "upper") %>%
  ggplot(aes(x = expenditure, y = Combined_SAT)) +
  geom_point() +
  geom_smooth(method = 'lm', formula = y ~ x) +
  theme_bw() +
  labs(
    title = "Distribution of SAT Scores",
    subtitle = "Specifically for states with >=50% students testing",
    x = "Expenditure",
    y = "Combined SAT Score"
  )

lower_plot <- grouped_SAT_data %>%
  filter(division == "lower") %>%
  ggplot(aes(x = expenditure, y = Combined_SAT)) +
  geom_point() +
  geom_smooth(method = 'lm', formula = y ~ x) +
  theme_bw() +
  labs(
    title = "Distribution of SAT Scores",
    subtitle = "Specifically for states with <50% students testing",
    x = "Expenditure",
    y = "Combined SAT Score"
  )

upper_plot
```




```
lower_plot
```




```r
#Using the coin data once again
#X1 column vector is the column containing number of pennies in each handful
#X2 column vector is the column containing number of coins in each handful
#Y column vector is the column containing the value of each handful
X1 <- coin_data$X..of.pennies
X2 <- coin_data$X..of.coins
Y <- coin_data$value.in.cents

#Creating (nxp) X matrix
X <- cbind(1,X1,X2)

#Finding the inverse of X-transpose times X
Xinv <- solve(t(X) %*% X)

#Multiplying the above by X-transpose and Y to get beta-hat
bhat <- Xinv %*% t(X) %*% Y
bhat
```

```
##         [,1]
##    -1.749313
## X1 -7.027829
## X2  8.402602
```

```r
#Finding the predicted y values using the predicted beta from above
yhat <- X %*% bhat

#Finding the residuals
resids <- Y - yhat

#Finding the number of observations
n <- nrow(coin_data)

#Finding the estimate of sigma
rmse <- sqrt(sum(resids^2)/(n-3))
rmse
```

```
## [1] 4.807188
```

```r
#Finding the standard error
SE <- rmse * sqrt(diag(Xinv))
SE
```

```
##                  X1        X2
## 2.1167077 0.6572669 0.5030123
```