Example: The Auto Data

`Auto` data of 9 variables about 392 car models in the 1980s.

The variables include

- `acceleration`: Time to accelerate from 0 to 60 mph (in seconds)
- `horsepower`: Engine horsepower
- `weight`: Vehicle weight (lbs.)

Description of all 9 variables: https://rdrr.io/cran/ISLR/man/Auto.html

You can download the data at

https://www.stat.uchicago.edu/~yibi/s224/data/Auto.txt

Please **change the working directory** to the folder where `Auto.txt` is stored, and load the data as follows.

```
Auto = read.table("Auto.txt", h=T)
```

How to Do Regression in R?

```
lm(acceleration ~ weight + horsepower, data=Auto)

Call:
lm(formula = acceleration ~ weight + horsepower, data = Auto)

Coefficients:
(Intercept)       weight   horsepower
    18.4358       0.0023      -0.0933
```

The `lm()` command above asks R to fit the model

$$\text{acceleration} = \beta_0 + \beta_1 \text{weight} + \beta_2 \text{horsepower} + \varepsilon$$

and R gives us the regression equation

$$\widehat{\text{acceleration}} = 18.4358 + 0.0023\,\text{weight} - 0.0933\,\text{horsepower}$$

More R Commands

```
lm1 = lm(acceleration ~ weight + horsepower, data=Auto)
lm1$coef         # show the estimated beta's
(Intercept)      weight  horsepower
  18.435791    0.002302   -0.093313
```

More R Commands

```
lm1 = lm(acceleration ~ weight + horsepower, data=Auto)
lm1$coef          # show the estimated beta's
(Intercept)      weight  horsepower
  18.435791    0.002302   -0.093313
```

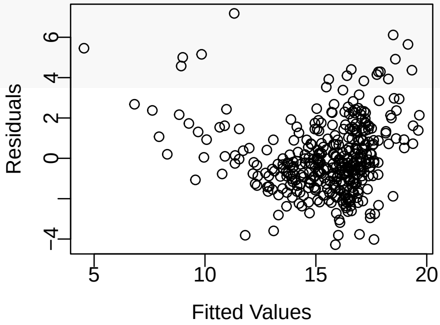
```
lm1$fit           # show the fitted values
lm1$res           # show the residuals
```

More R Commands

```
lm1 = lm(acceleration ~ weight + horsepower, data=Auto)
lm1$coef         # show the estimated beta's
(Intercept)      weight  horsepower
  18.435791    0.002302   -0.093313
```

```
lm1$fit          # show the fitted values
lm1$res          # show the residuals
```

```
plot(lm1$fit,lm1$res,
     xlab="Fitted Values",
     ylab="Residuals")
```



Interpretation of Regression Coefficients

Interpretation of the Intercept β_0

β_0 = intercept = the mean value of Y when all X_j' are 0.

- may have no practical meaning
  e.g., β_0 is meaningless in the `Auto` model as no car has 0 weight

Interpretation of the regression coefficient for β_j

β_j = the regression coefficient for X_j, is the mean change in the response Y when X_j is increased by one unit **holding other X_i's constant**.

- Also called the **partial regression coefficients** because they are *adjusted for the other covariates*
- Interpretation of β_j depends on the presence of other predictors in the model
  e.g., the 2 β_1's in the 2 models below have different interpretations

$$\text{Model } 1: \; Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$
$$\text{Model } 2: \; Y = \beta_0 + \beta_1 X_1 + \varepsilon.$$

Something Wrong?

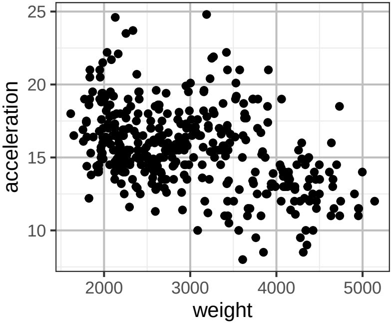
```
# Model 1
lm(acceleration ~ weight, data=Auto)$coef
(Intercept)      weight
  19.572666   -0.001354
# Model 2
lm(acceleration ~ weight + horsepower, data=Auto)$coef
(Intercept)      weight  horsepower
  18.435791    0.002302   -0.093313
```

The coefficient $\widehat{\beta}_1$ for weight is *negative* in the Model 1 but *positive* in the Model 2.

Do heavier cars require more or less time to accelerate from 0 to 60 mph?

Effect of `weight` Not Controlling for Other Predictors

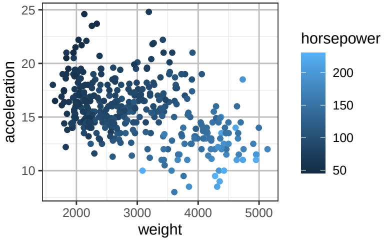
```
library(ggplot2)
ggplot(Auto, aes(x=weight, y=acceleration)) + geom_point()
```

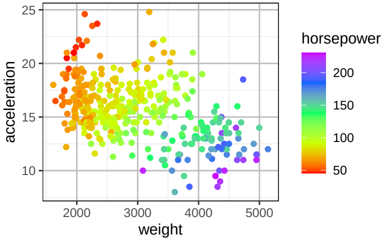


From the scatter plot above, are `weight` and `acceleration` are positively or negatively associated? Do heavier vehicles generally require more or less time to accelerate from 0 to 60 mph? Is that reasonable?

Effect of weight Controlling for horsepower (1)

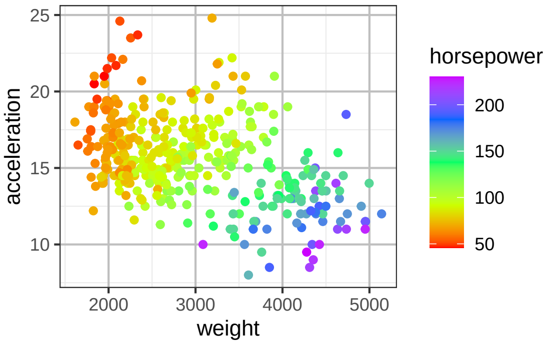
```
ggplot(Auto, aes(x=weight, y=acceleration, col=horsepower)) +
  geom_point()
ggplot(Auto, aes(x=weight, y=acceleration, col=horsepower)) +
  geom_point() + scale_color_gradientn(colours = rainbow(5))
```





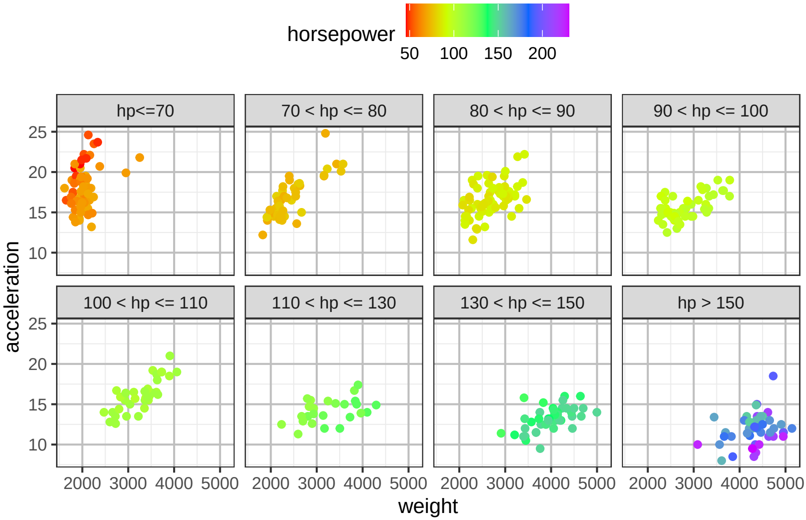
Effect of weight Controlling for horsepower (2)

```
ggplot(Auto, aes(x=weight, y=acceleration, col=horsepower)) +
  geom_point() + scale_color_gradientn(colours = rainbow(5))
```



Consider car models of similar `horsepower` (similar color), are `weight` and `acceleration` positively or negatively correlated?

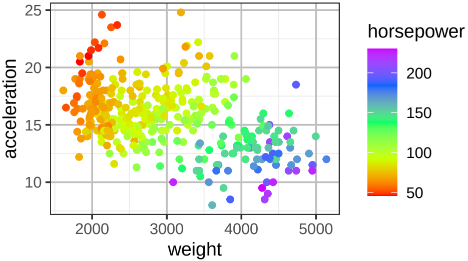
Effect of `weight` Controlling for `horsepower` (3)



R codes for the plot on the previous page

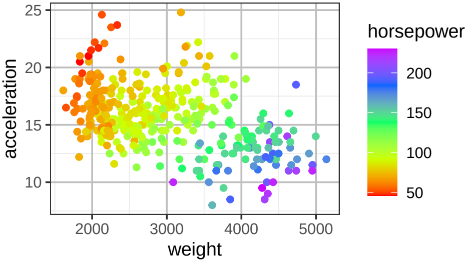
```
Auto$hp = cut(Auto$horsepower,
              breaks=c(45,70, 80, 90,100,110, 130, 150,230),
              labels=c("hp<=70", "70 < hp <= 80", "80 < hp <= 90",
                       "90 < hp <= 100", "100 < hp <= 110",
                       "110 < hp <= 130",
                       "130 < hp <= 150", "hp > 150"))
ggplot(Auto, aes(x=weight, y=acceleration, col=horsepower)) +
  geom_point() + scale_color_gradientn(colours = rainbow(5)) +
  facet_wrap(~hp, nrow=2) + theme(legend.position="top")
```

Example: Auto Data — Simpson's Paradox



Why is the association btw `acceleration` and `weight` flipped from positive to negative when `horsepower` is ignored?

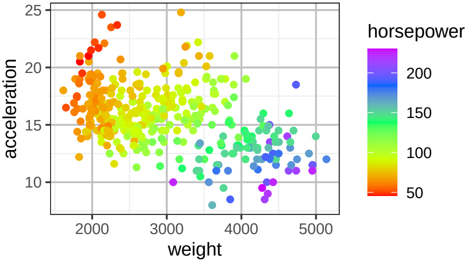
Example: Auto Data — Simpson's Paradox



Why is the association btw `acceleration` and `weight` flipped from positive to negative when `horsepower` is ignored?

- Heavier vehicles (purple dots) tend to have more horsepower while lighter ones (red dots) tend to have less

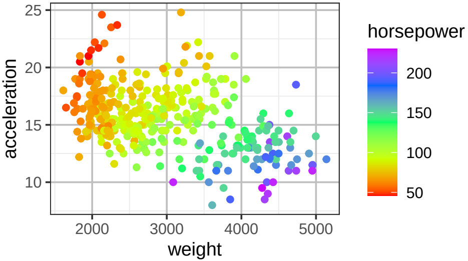
Example: Auto Data — Simpson's Paradox



Why is the association btw `acceleration` and `weight` flipped from positive to negative when `horsepower` is ignored?

- Heavier vehicles (purple dots) tend to have more horsepower while lighter ones (red dots) tend to have less
- Vehicles with more horsepower (purple dots) require less time to accelerate while those with less (red dots) require more

Example: Auto Data — Simpson's Paradox



Why is the association btw `acceleration` and `weight` flipped from positive to negative when `horsepower` is ignored?

- Heavier vehicles (purple dots) tend to have more horsepower while lighter ones (red dots) tend to have less
- Vehicles with more horsepower (purple dots) require less time to accelerate while those with less (red dots) require more
- Hence, when ignoring `horsepower`, it looks like heavier vehicles require less time to accelerate, though heavier vehicles require more time to accelerate after the effect of `horsepower` is adjusted (which means considering only vehicles with similar `horsepower`)

What We Mean by “Adjusted for Other Coveriates”?

For a multiple linear regression model with p predictors

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

β_j represents the effect of X_j on the respone variable Y after it has been **adjusted** for all of $X_1, \ldots, X_p$ except X_j.

What does “adjusted for” mean?

What We Mean by "Adjusted for Other Coveriates" (2)?

The LS estimate $\widehat{\beta}_j$ for β_j in the MLR model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

would be identical to the slope for the SLR model computed as follows.

1. Regress Y on all other X_k's except X_j
2. Regress X_j on all other X_k's except X_j
3. Fit a SLR model using the residuals from Step 1 as the response and the residuals from Step 2 as the predictor.

What We Mean by “Adjusted for Other Coveriates” (2)?

The LS estimate $\widehat{\beta}_j$ for β_j in the MLR model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

would be identical to the slope for the SLR model computed as follows.

1. Regress Y on all other X_k’s except X_j
2. Regress X_j on all other X_k’s except X_j
3. Fit a SLR model using the residuals from Step 1 as the response and the residuals from Step 2 as the predictor.

Moreover, the intercept obtained in Step 3 would be 0.

What We Mean by “Adjusted for Other Coveriates” (2)?

The LS estimate $\widehat{\beta}_j$ for β_j in the MLR model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

would be identical to the slope for the SLR model computed as follows.

1. Regress Y on all other X_k's except X_j
2. Regress X_j on all other X_k's except X_j
3. Fit a SLR model using the residuals from Step 1 as the response and the residuals from Step 2 as the predictor.

Moreover, the intercept obtained in Step 3 would be 0.

This proof of this result involves complicated matrix algebra and hence is omitted. We just illustrate with an example.

For the Auto Data, recall we have fit the model

$$\text{acceleration} = \beta_0 + \beta_1 \text{weight} + \beta_2 \text{horsepower} + \varepsilon$$

and obtained the estimate for β_1 to be $\widehat{\beta_2} = 0.0023$.

Step 1. Regress `acceleration` on `horsepower`. Let `RY` be the residuals of this model.

```
RY = lm(acceleration ~ horsepower, data=Auto)$res
```

Step 2. Regress `weight` on `horsepower`. Let `RWT` be the residuals of this model.

```
RWT = lm(weight ~ horsepower, data=Auto)$res
```

Step 3. Regress `RY` on `RWT`.

```
lm(RY ~ RWT)$coef
(Intercept)         RWT
  7.352e-17   2.302e-03
```

Observe that

- the **estimated intercept is exactly 0** (slightly off due to rounding error)
- the estimated coefficient for `RWT` is *exactly same* estimated coefficient for `weight` in the model.

```
lm(acceleration ~ weight + horsepower, data=Auto)$coef
(Intercept)      weight  horsepower
  18.435791    0.002302   -0.093313
```

$$
\begin{aligned}
RY &= \text{acceleration} - \tilde{\beta}_0 - \tilde{\beta}_1 \text{horsepower} \\
&= \text{the part of } \texttt{acceleration} \text{ not explained by } \texttt{horsepower}
\end{aligned}
$$

`weight` might be correlated with other predictors in the model.

$$\text{weight} = \check{\beta}_0 + \check{\beta}_1 \text{horsepower} + \text{error}$$

We can break `weight` into 2 components:

- a part that's linear w/ of `horsepower`, and
- the part `RWT` that is uncorrelated with `horsepower`

The first part is useless in predicting `acceleration` since `horsepower` haS been included in the model. Only `RWT` provides the additional information that `horsepower` cannot provide.