Regression & Correlation

Correlations

1. simple correlation

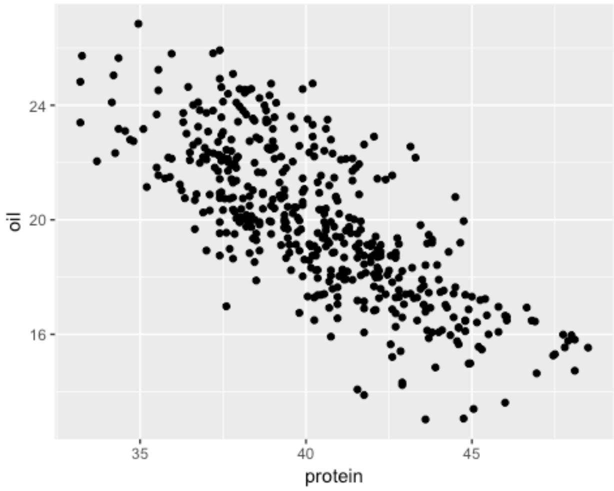
หลาย ๆ ครั้ง เวลาที่เราต้องการศึกษาความสัมพันธ์ระหว่าง 2 ตัวแปร เรามักจะไม่ทราบว่า ตัวไหนเป็นสาเหตุ ตัวไหนเป็นผลลัพธ์ แต่เรายังอยากรู้ว่า 2 ตัวแปรนั้นมีความสัมพันธ์กันหรือไม่ ซึ่งเราก็ยังสามารถทราบได้ โดยการหาสหสัมพันธ์ (correlation)

เราลองมาดูข้อมูล Australian Soybean ของเรากันอีกครั้งหนึ่ง

```
## # A tibble: 464 x 10
##    env   loc    year gen   yield height lodging  size protein   oil
##    <chr> <chr> <dbl> <chr> <dbl>  <dbl>   <dbl> <dbl>   <dbl> <dbl>
##  1 L70   Lawes  1970 G01    2.39   1.44    4.25  8.45    36.7  20.9
##  2 L70   Lawes  1970 G02    2.28   1.45    4.25  9.95    37.6  20.7
## # …
```

ในตารางข้อมูลนี้เรามีข้อมูลเกี่ยวกับคุณภาพของถั่วเหลืองหลายอย่าง เช่น `size` `oil` และ `protein` เราจะเริ่มดูจากความสัมพันธ์ระหว่าง protein กับ oil ก่อน วิธีที่ดีที่สุด คือ เริ่มจากการวาดกราฟให้ตัวเองเห็นภาพก่อน

```
ggplot(soy, aes(x = protein, y = oil)) +
  geom_point()
```



ดูจากกราฟแล้วก็น่าจะมีความสัมพันธ์กัน เราสามารถตรวจสอบความสัมพันธ์ของ oil และ protein อย่างเป็นทางการด้วยการใช้ฟังก์ชัน `cor.test`

```
cor.test(soy$protein, soy$oil)

##
##  Pearson's product-moment correlation
##
## data:  soy$protein and soy$oil
## t = -24.956, df = 462, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7939767 -0.7160695
## sample estimates:
##       cor
## -0.75771
```

มีสองอย่างที่สำคัญจากผลทดสอบนี้ คือ

1) P-values: บอกว่าความสัมพันธ์นี้มีนัยสำคัญหรือไม่ (ความสัมพันธ์นี้มีสัมประสิทธิ์ไม่เป็น 0 จริงไหม ถ้าต่างจริง P จะต้อง**น้อยกว่าหรือเท่ากับ** 0.05
2) cor: คือ **สัมประสิทธิ์สหสัมพันธ์ (correlation coefficient)** เป็นตัวบอกความรุนแรงของความสัมพันธ์นั้น ๆ กล่าวคือ
    - cor = 0 หมายถึง ไม่มีความสัมพันธ์ใด ๆ
    - cor = 1 คือ มีความสัมพันธ์เชิง**บวก** (**positive** relationship) อย่างรุนแรงที่สุด (x เพิ่ม y เพิ่ม)
    - cor = -1 คือ มีความสัมพันธ์เชิง**ลบ** (**negative** relationship) อย่างรุนแรงที่สุด (x เพิ่ม y ลด)

ในกรณีนี้เรามีความสัมพันธ์เชิงลบที่มีนัยสำคัญ (cor = -0.75, P < 0.001) ระหว่าง `protein` และ `oil` ซึ่งหมายความว่าปริมาณน้ำมันมีแนวโน้มที่จะลดลงเมื่อปริมาณโปรตีนเพิ่มขึ้น

2. Correlation Matrix

ในโลกที่เต็มไปด้วยข้อมูลอย่างทุกวันนี้ เราอาจจะไม่มีเวลามานั่งดูทีละตัวแปรที่ละคู่ เรามีวิธีที่จะพิจารณาตัวแปรทุกคู่พร้อม ๆ กันในเวลาเดียวกันดังนี้

เริ่มแรกจากเลือกข้อมูลแค่บางส่วนมาวิเคราะห์ด้วยฟังก์ชัน `select`

```
soy2 <- soy %>%
        select(height, yield, size, protein, oil)
```

หลังจากนั้นเราสามารถใช้คำสั่ง `rcorr` จาก package `Hmisc` เพื่อวิเคราะห์หลาย ๆ ตัวแปรพร้อมกันในทีเดียว

```
library(Hmisc)

rcorr(as.matrix(soy2))

##           height yield  size protein   oil
## height      1.00 -0.22 -0.52    0.28 -0.50
## yield      -0.22  1.00  0.51   -0.34  0.54
## size       -0.52  0.51  1.00   -0.41  0.76
## protein     0.28 -0.34 -0.41    1.00 -0.76
## oil        -0.50  0.54  0.76   -0.76  1.00
##
## n= 464
##
##
## P
##           height yield size protein oil
## height           0     0    0       0
## yield      0           0    0       0
## size       0      0         0       0
## protein    0      0     0           0
## oil        0      0     0    0
```

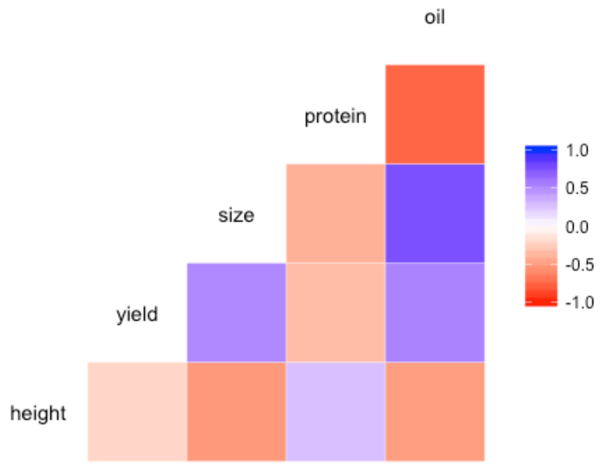
ตัวเลขใน matrix ด้านบนคือค่าสัมประสิทธิ์ของสหสัมพันธ์ ส่วน matrix ด้านล่างแสดงค่า P-value ของความสัมพันธ์นั้น ๆ

ตัวอย่างเช่น

ซึ่งอาจจะดูยากสักหน่อยเราสามารถดูความสัมพันธ์ได้ง่ายขึ้นในรูปกราฟโดย `GGally` และ `corrplot`

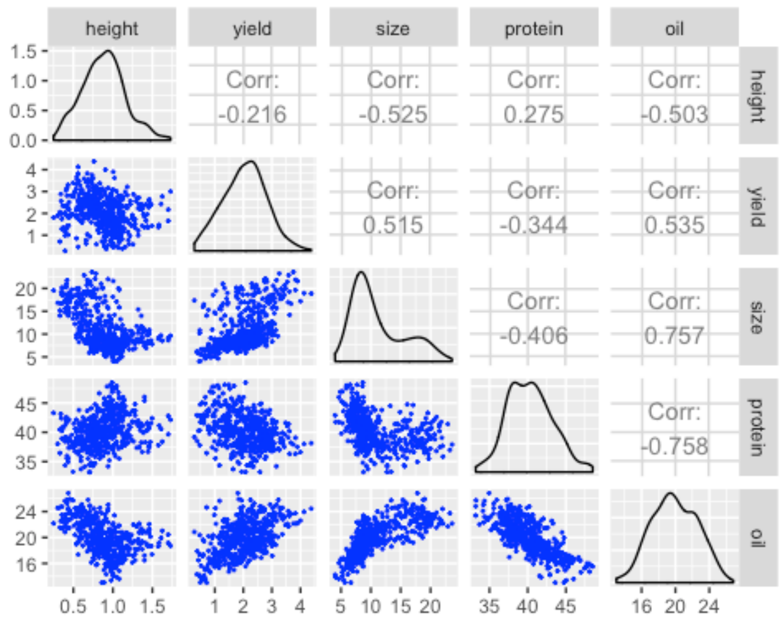
<u>**วิธีที่** 1</u> `ggcorr` แสดงเฉพาะสีตามความสัมพันธ์จากค่าสัมประสิทธิ์

```
library(GGally)
ggcorr(soy2, low = "red", mid = "white", high = "blue")
```



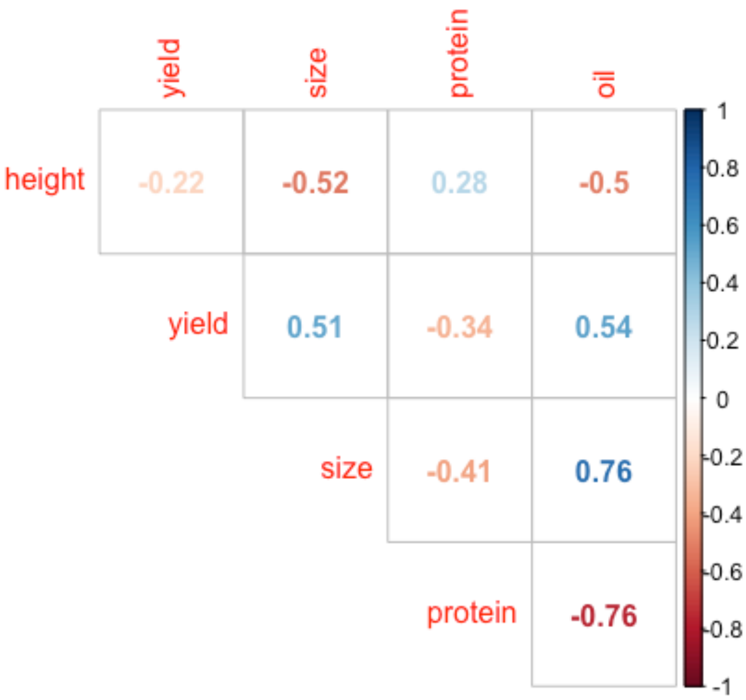
<u>**วิธีที่** 2</u> `ggpairs` จะแสดงทั้งข้อมูลดิบ การกระจายของข้อมูล และ ค่าสัมประสิทธิ์ความสัมพันธ์เป็นตัวเลข

```
ggpairs(soy2, lower = list(continuous = wrap("points", size = 0.3, color =
"blue")))
```

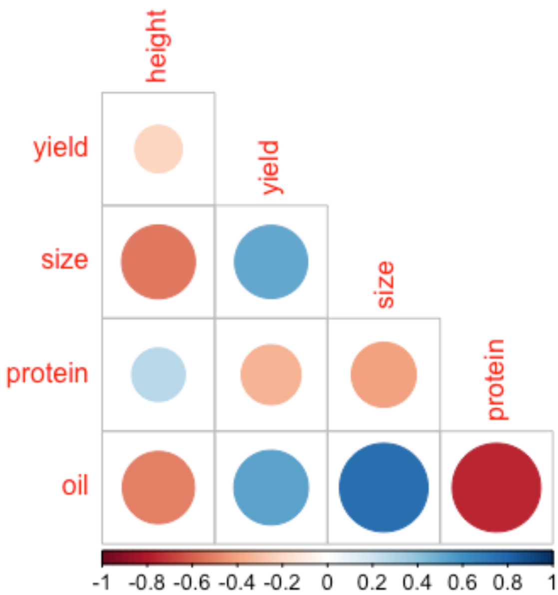


<u>วิธีที่ 3</u> `corrplot` สามารถแสดงผลสัมประสิทธิ์สหสัมพันธ์ได้หลายวิธีเช่น .....

```
library(corrplot)
cor.soy <- cor(soy2)
corrplot(cor.soy, type = "upper", method = "number", diag = F)
```




```
corrplot(cor.soy, type = "lower", method = "circle", diag = F)
```



Regression: Basics

การวิเคราะห์สมการถดถอย (regression) คือการหาความสัมพันธ์ระหว่างตัวแปรต้น (independent variable) และ ตัวแปรตาม (dependent variable) ที่เป็นตัวแปรต่อเนื่องทั้งสองตัวแปร โดยเราสนใจว่าตัวแปรต้นมีผลต่อตัวแปรตามมากน้อยเพียงไร โดยวันนี้จะได้มาทำความเข้าใจเกี่ยวกับวิธีการทางสถิติที่ใช้อธิบายความสัมพันธ์เหล่านี้กัน

1. รู้จักข้อมูลของเรา

เราจะใช้ข้อมูลจากการทดลองเกี่ยวกับการเพิ่มน้ำหนักของหมูซึ่งมีหน้าตาแบบนี้

```
## # A tibble: 50 x 5
##    treatment rep    feed weight1  gain
##    <chr>     <chr> <dbl>   <dbl> <dbl>
##  1 T1        R1      674      30   165
##  2 T1        R2      628      21   156
##  3 T1        R3      661      21   159
##  4 T1        R4      694      33   167
##  5 T1        R5      713      27   170
##  6 T1        R6      585      24   146
##  7 T1        R7      575      20   130
##  8 T1        R8      638      29   151
##  9 T1        R9      632      28   164
## 10 T1        R10     637      26   158
## # … with 40 more rows
```

โดยมีคอลัมน์ต่อไปนี้

- `treatment` คือ ชนิดของอาหารที่ให้กับหมูแต่ละตัว
- `rep` คือ แต่ละซ้ำ (replication) ก็คือหมูแต่ละตัวนั่นเอง
- `feed` คือ ปริมาณอาหารที่หมูแต่ละตัวกินเข้าไประหว่างการทดลอง
- `weight1` คือ น้ำหนักของหมูก่อนเริ่มต้นการทดลอง
- `gain` คือ น้ำหนักของหมูที่**<u>เพิ่มขึ้น</u>**หลังจากการทดลอง

เราจะลองมาวิเคราะห์ความสัมพันธ์ระหว่างปริมาณอาหาร (feed) และ น้ำหนักที่เพิ่มขึ้น (gain) เราคิดว่า....

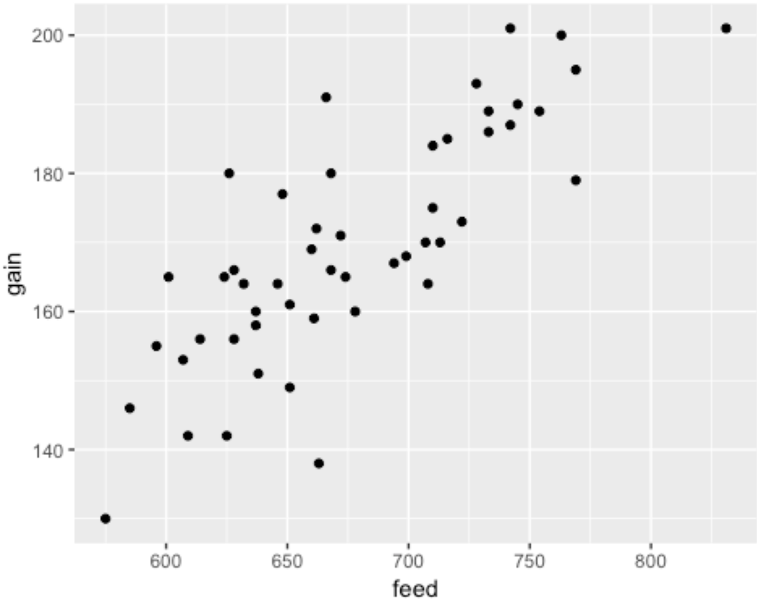
ตัวแปรต้น คือ ....

ตัวแปรตาม คือ ...

สมมติฐานก็คือ ....

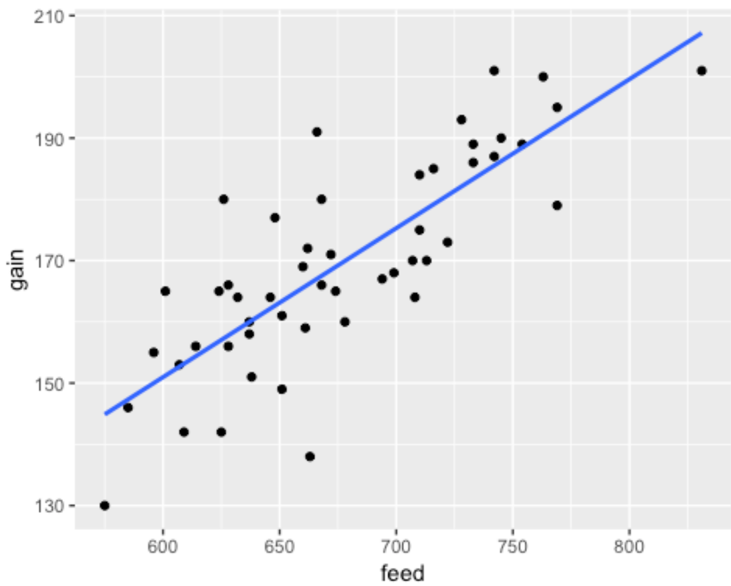
2. สร้างกราฟดูรูปแบบก่อน

เช่นเคย สิ่งแรกที่เราจะต้องทำคือการสร้างกราฟเพื่อแสดงความสัมพันธ์ระหว่างตัวแปรทั้งสองดังนี้



แนวโน้มเป็นอย่างไร ปริมาณอาหารมีผลต่อน้ำหนักจริงหรือไม่

เราลองใส่เส้นแนวโน้ม (trend line) แบบเส้นตรงลงไป ด้วยฟังก์ชั่น `geom_smooth`



สิ่งที่เราทำอยู่นี้คือการใส่สมการเส้นตรง (linear equation/linear model) ลงไปในข้อมูลของเราเพื่ออธิบายข้อมูล

เราคิดว่าสมการเส้นตรงนี้อธิบายข้อมูลเราดีพอไหม?

3. สร้างสมการเส้นตรงอธิบายข้อมูลดีกว่า

เราจะใช้ฟังก์ชัน `lm` เพื่อสร้างสมการเส้นตรงเพื่ออธิบายข้อมูลของเราดังนี้

```
pig.lm <- lm(gain ~ feed, data = pig)
pig.lm


## Call:
## lm(formula = gain ~ feed, data = pig)
##
## Coefficients:
## (Intercept)         feed
##      5.0319       0.2432
```

สมการที่เราสร้างนี้ คือ สมการที่น้ำหนักที่เพิ่มขึ้น (`gain`) เป็นผลมาจากอาหารที่กินเข้าไป (`feed`) ซึ่งมีจุดตัดแกน y (`intercept`) และ ความชัน (`slope`) หรือ `feed` ใน `output` นี้ ซึ่งเขียนเป็นสมการง่ายๆได้แบบนี้

gain = intercept + slope x gain = 5.0319 + 0.2432 x gain

ซึ่งสามารถเขียนเป็นรูปสมการอย่างเป็นทางการได้แบบนี้

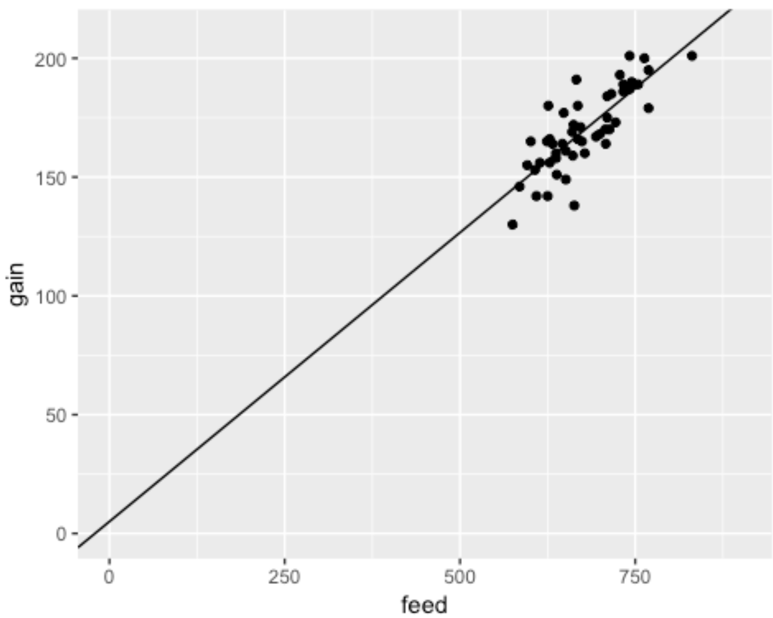
$$\hat{y}_i = \alpha + \beta \times \mathrm{x}_i$$

โดยที่

$\hat{y}_i$ = ค่า y ที่ได้จากสมการ α = จุดตัดแกน y

β = ความชัน x_i = ค่า x ของแต่ละค่าในสมการ

ซึ่งถ้าเรานำข้อมูลข้างต้นมาวาดเป็นกราฟเส้นตรง จะได้เป็นกราฟแบบนี้



3.1 จุดตัด (intercept) คืออะไร

ลองกลับไปดูที่สมการอีกครั้ง

$$\hat{y}_i = \alpha + \beta \times \mathrm{x}_i$$

ถ้าเราต้องการทราบว่า α หรือจุดตัดแกน y คืออะไร เราจะต้องทำให้พจน์ที่สองเป็นศูนย์ ซึ่งวิธีที่ง่ายที่สุดก็คือการทำให้ x_i เป็น 0 จึงได้ออกมาแบบนี้

$$\hat{y}_i = \alpha$$

แปลง่ายๆก็คือ ค่า y ที่จะได้เมื่อค่า x เป็น 0 หรือในบริบทของข้อมูลของเรา ก็คือ น้ำหนักของหมูที่เพิ่มขึ้น (gain) เมื่อไม่ให้อาหารอะไรเลย (feed = 0)

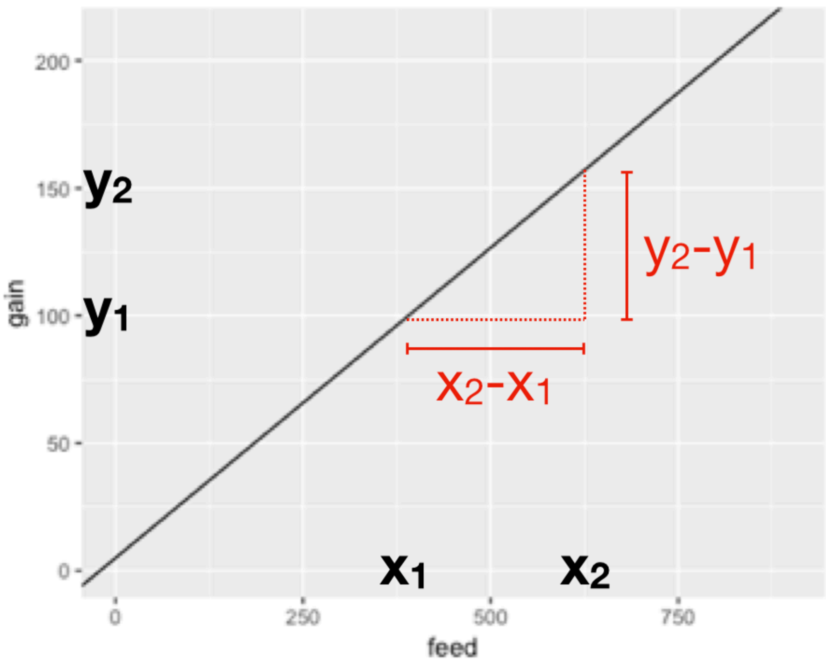
เราคิดว่าเป็นไปได้ไหมที่หมูจะมีน้ำหนักเพิ่มเมื่อไม่ให้อาหารอะไรเลย

3.2 ความชัน (slope) คืออะไร

Slope หรือ ความชัน คือ การเพิ่มขึ้นของค่า y เมื่อค่า x เพิ่มขึ้น โดยสามารถเขียนสูตรคำนวณได้อย่างเป็นทางการได้แบบนี้:

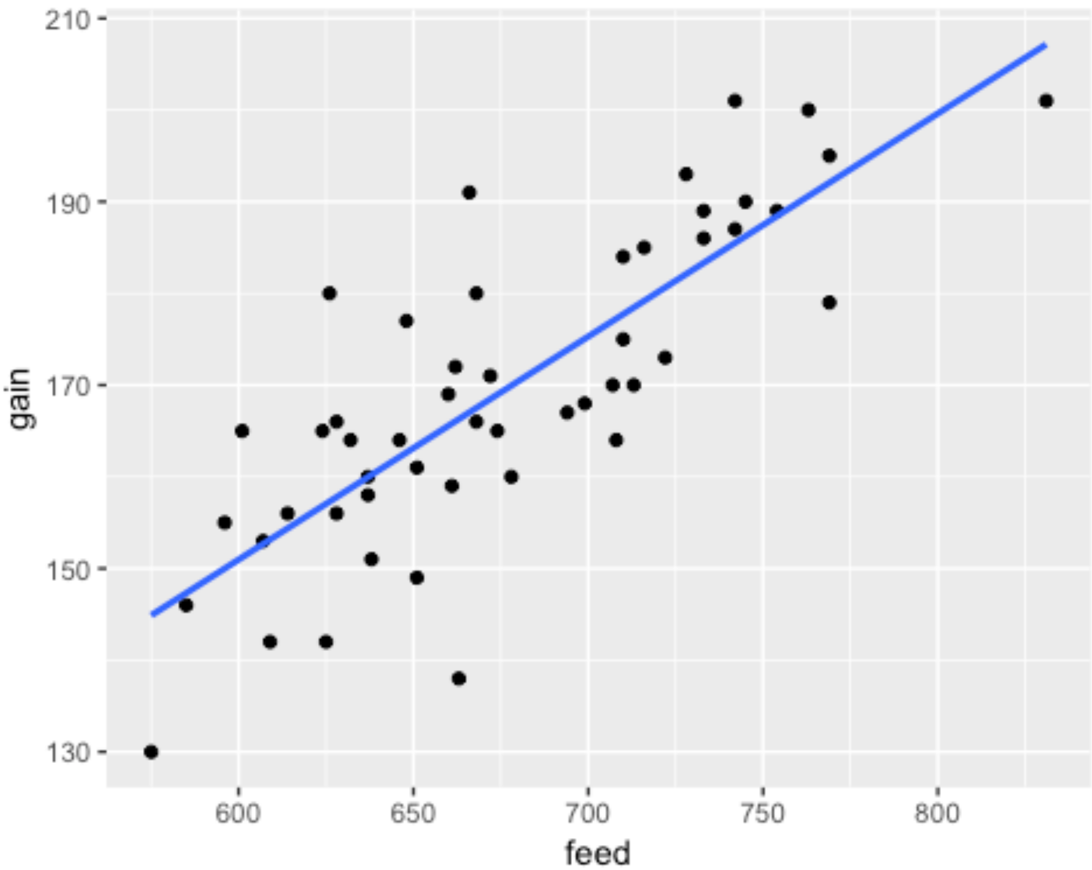
$$\text{slope} = \frac{\hat{y}_2 - \hat{y}_1}{\mathrm{x}_2 - \mathrm{x}_1}$$

ในบริบทของข้อมูลนี้ ก็คือ น้ำหนักของหมูที่เพิ่มขึ้นเมื่อเพิ่มอาหารขึ้นมา ถ้าชันมากขึ้น แปลว่าเพิ่มปริมาณอาหารเท่ากัน น้ำหนักสามารถเพิ่มขึ้นได้เร็วขึ้นกว่าเดิม ในอีกมุมก็คือว่า ปริมาณอาหารมีผลต่อน้ำหนักเพิ่มขึ้นเท่าไหร่



4. ข้อมูลของเราฟิตกับโมเดลดีแค่ไหน

ลองกลับมาดูกราฟอีกรอบหนึ่ง



จะเห็นได้ว่าข้อมูลของเราไม่ฟิตเปรี๊ยะกับเส้น regression ที่เราสร้างขึ้น เพราะข้อมูลที่ไหนในโลกจะ perfect ขนาดนั้นก็คงไม่มี ดังนั้น **ตัวข้อมูล** ของเราจะประพฤติตัวคล้ายกับสมการเส้นตรง แต่ไม่ใช่เส้นตรงเสียทีเดียว ตามนี้

$$y_i = \alpha + \beta \times x_i + \epsilon_i$$

y_i และ x_i ที่ไม่มีหมวก หมายถึง ค่าข้อมูลจริง ๆ

ϵ_i หมายถึง ค่าความคลาดเคลื่อนของแต่ละจุดข้อมูล ที่เป็นผลมาจากตัวแปรอื่น ๆ ที่ไม่ใช่ตัวแปรที่เราสนใจ

ในกรณีของเราที่พิจารณาผลของปริมาณอาหารที่ให้ (feed) เป็นตัวแปรเดียว จะทำให้เราอธิบายการเพิ่มขึ้นของน้ำหนักของหมูได้เพียงบางส่วนเท่านั้น เพราะว่าปริมาณอาหารที่ให้ ไม่ใช่เพียงปัจจัยเดียวที่ทำให้น้ำหนักหมูขึ้น อาจจะมีปัจจัยอื่น ๆ ด้วย จุดข้อมูล จึงไม่ได้อยู่บนเส้นสมการเส้นตรงทั้งหมด

ถ้าความคาดเคลื่อนมาก จุดข้อมูลก็จะกระจายออกจากเส้นมากเท่านั้น ลองไปทดสอบความเข้าใจเราเกี่ยวกับความคาดเคลื่อนและความชันในโมเดล ได้ที่ https://ekraichak.shinyapps.io/regression/

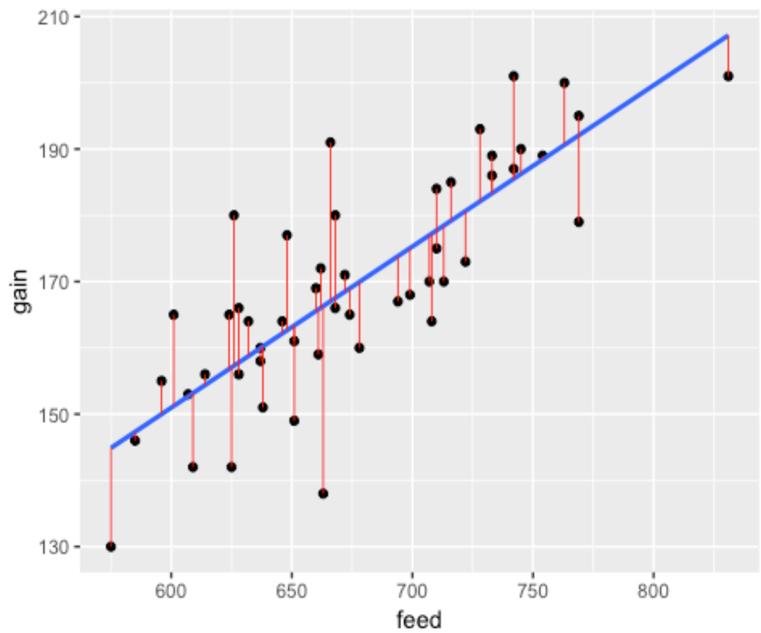
5. Regression ทดสอบอะไร

5.1 สมการเส้นตรงของเราทำงานได้ดีแค่ไหน?

เราสามารถวัดความสามารถของสมการเส้นตรงในการอธิบายชุดข้อมูลของเราได้ โดยวัดว่า **จุดข้อมูลนั้นอยู่ห่างหรือแตกต่างจากเส้นสมการมากเท่าไหร่** เราเรียกความแตกต่างนี้ว่า **residuals** ซึ่งสามารถคำนวณได้ดังนี้

$$\text{residuals} = y_i - \hat{y}_i = y_i - (\alpha + \beta \times \mathrm{x}_i)$$

$\hat{y}_i$ ก็คือค่าของ y ที่ได้จากสมการในการพยายามเอาเส้นตรงมาอธิบายข้อมูลของเรา ซึ่งก็ได้มาจากสมการเส้นตรงปกติก็คือ $\alpha + \beta \times \hat{x}_i$ โดยเราสามารถมองเห็น residual ได้ตามภาพนี้



สมการหรือโมเดลที่อธิบายข้อมูลได้ดี ก็คือ สมการที่มีค่า residuals น้อยมากที่สุดเท่าที่เป็นไปได้ ซึ่งในกรณีของเราก็เหมือนจะมีความแตกต่างระหว่างจุดกับตัวเส้นอยู่พอสมควร แล้วเมื่อไหร่ที่เราจะเรียกว่าน้อยพอ เราจึงต้องมีวิธีการคำนวณ residuals ทั้งหมด หรือการคำนวณค่าที่เรียกว่า **residual sum of square** ($SS_{residuals}$)

$$SS_{residuals} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

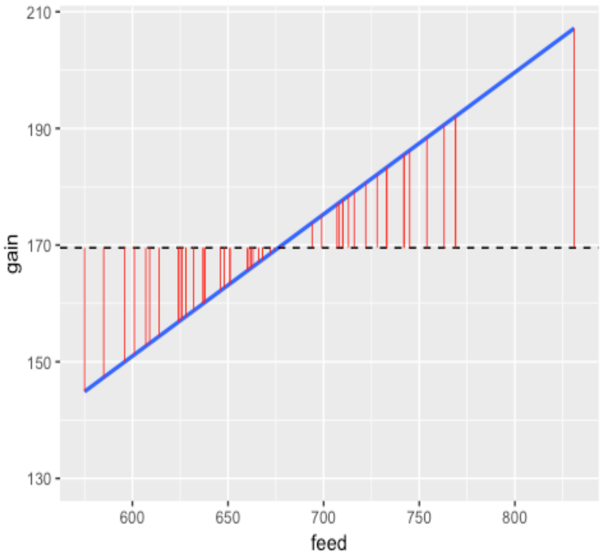
ซึ่งค่านี้ไม่ใช่อะไรพิเศษ ก็แค่ความแตกต่างระหว่างจุดข้อมูล (y_i) และค่าที่สมการให้มา ($\hat{y}_i$) แล้วนำไปยกกำลังสอง เพื่อให้ทั้งบวกและลบมีเครื่องหมายเป็นบวกทั้งหมด แล้วจึงมาจับรวมกัน **หน้าที่หลักของการวิเคราะห์ regression คือทำให้สมการนั้นมีค่า $SS_{residuals}$ น้อยที่สุด** (ซึ่งมีวิธีการ proof ค่อนข้างจะยากเกินขอบเขตของวิชานี้ เอาเป็นว่ามีวิธีการละกัน) เราเรียกวิธีการนี้ว่า Ordinary Least Square (OLS) Estimation

แล้วเราจะรู้ได้อย่างไรว่า $SS_{residuals}$ ที่เราคำนวณได้มันมากหรือน้อยอย่างไร เราจึงต้องมีตัวเทียบโดยการคำนวณ sum of square อีกสองแบบได้แก่

1. regression sum of square ($SS_{regression}$)

$$SS_{regression} = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$$

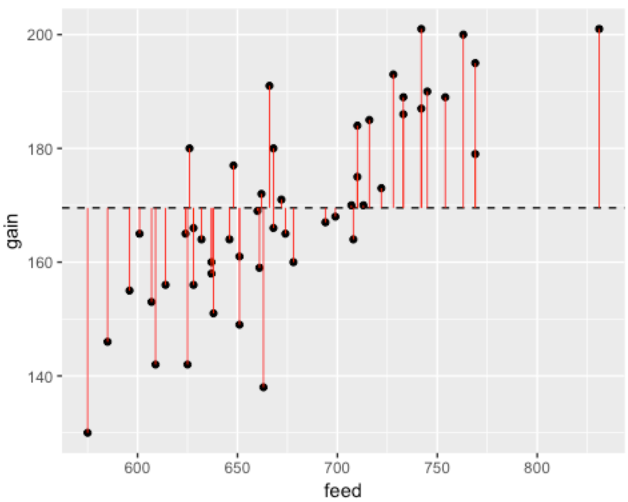
อันนี้เป็น sum of square **<u>ของตัวเส้น regression</u>** ว่าแตกต่างจากค่าเฉลี่ยของ y ($\bar{y}$) การคำนวณนี้ เป็นการดูว่าเส้น regression ที่วาดมานี้แตกต่างจากที่ไม่มีความสัมพันธ์เลยอย่างไร (เส้นแนวนอน ที่ทุกค่า x ค่า y เท่ากับค่าเฉลี่ยของ y)



2. total sum fo square (SS_{total})

$$SS_{total} = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

อันนี้เป็น sum of square ของ**<u>ตัวจุดข้อมูล</u>** ว่าแตกต่างจากค่าเฉลี่ยของ y ($\bar{y}$) การคำนวณนี้ เป็นการดูว่า จุดข้อมูลนี้เป็นอย่างไรเมื่อเทียบกับไม่มีความสัมพันธ์เลยอย่างไร และด้วยความที่ sum of square นี้เป็นส่วนที่ความคาดเคลื่อนมากที่สุดจึงมีคุณสมบัติคือ $SS_{total} = SS_{residuals} + SS_{regression}$



ในการตัดสินใจว่าสมการของเราทำงานดีแค่ไหน เราจะคำนวณ the *coefficient of determinant* หรือ R^2 ด้วยสูตรนี้

$$R^2 = \frac{SS_{regression}}{SS_{total}} = 1 - \frac{SS_{residuals}}{SS_{total}}$$

ดังนั้น R^2 = 1 เป็นโมเดลที่สุดยอด อธิบายทุกอย่างในข้อมูลเรา, R^2 = 0 เป็นโมเดลที่แย่มาก (ไม่อธิบายอะไรเลย $SS_{residuals} = SS_{total}$)

เราสามารถคำนวณ sum of square ด้วยฟังก์ชัน **anova()**.

```
## Analysis of Variance Table
##
## Response: gain
##           Df Sum Sq Mean Sq F value    Pr(>F)
## feed       1 9131.0  9131.0  87.949 1.985e-12 ***
## Residuals 48 4983.4   103.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ในกรณีนี้, $SS_{regression}$ คือ `Sum sq` สำหรับ feed ดังนั้น SS_{total} ก็คือผลรวมของคอลัมน์ `Sum sq` ดังนั้นเราก็จะสามารถคำนวณ R^2 ได้ดังนี้

$$R^2 = \frac{9131.0}{9131.0 + 4983.4} = 0.64$$

หรืออธิบายแบบง่าย ๆ ว่า ตัวแปร **feed** อธิบายประมาณ 64% ของข้อมูลของเรา

5.2 ความชันของเราต่างจาก 0 หรือเปล่า?

ถ้า ความชัน หรือ slope (β) ของสมการ regression เป็น 0 นั่นแปลว่าไม่มีความสัมพันธ์ระหว่างตัวแปร x และ ตัวแปร y นักสถิติจึงคิดวิธีตรวจสอบสมมติฐานนี้โดยยืมวิธีการมาจาก t-test คือ

$$t = \frac{Observed - NoRelationship}{StandardError(SE)}$$

ในกรณีของ regression "No relationship" หมายความว่า slope (β) = 0 ดังนั้น t-statistics จึงเหลือแค่

$$t = \frac{\beta_{observed}}{SE}$$

ซึ่งจะนำไปคำนวณหาค่า P-value เหมือนกับ t-test ปกติ

6. ทำความเข้าใจกับ output ที่เราได้มาจาก regression

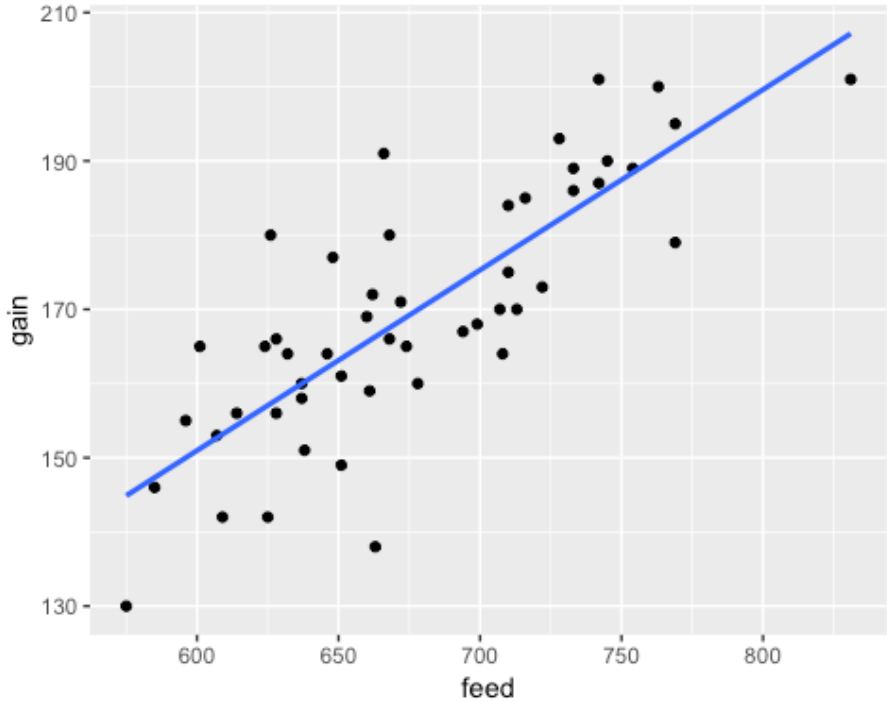
เราสามารถดูผลสรุปของ regression ได้จากฟังก์ชัน `summary()`

```
##
## Call:
## lm(formula = gain ~ feed, data = pig)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.291  -6.952   0.453   5.785  23.980




## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.03188   17.60076   0.286    0.776
## feed         0.24323    0.02594   9.378 1.98e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1




## Residual standard error: 10.19 on 48 degrees of freedom
## Multiple R-squared:  0.6469, Adjusted R-squared:  0.6396
## F-statistic: 87.95 on 1 and 48 DF,  p-value: 1.985e-12
```

7. สรุปวิธีการทำ regression แบบรวบรัด

Step 1 วาดกราฟเพื่อทราบ pattern

```
ggplot(pig, aes(x = feed, y = gain)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```



Step 2 สร้าง linear model

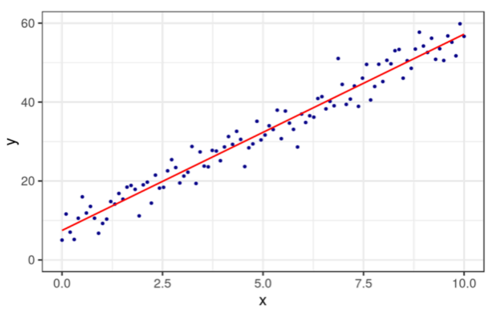
```
pig.lm <- lm(gain ~ feed, data = pig)
pig.lm

##
## Call:
## lm(formula = gain ~ feed, data = pig)
##
## Coefficients:
## (Intercept)         feed
##      5.0319       0.2432
```

Step 3 ใช้ summary ดูตัว model

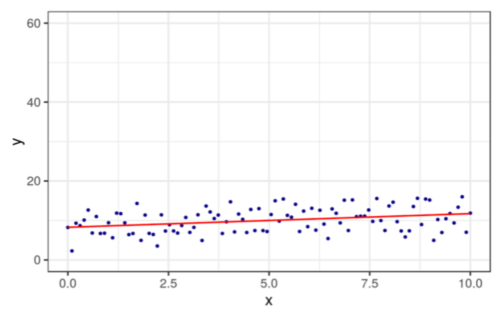
```
summary(pig.lm)

##
## Call:
## lm(formula = gain ~ feed, data = pig)
##
```

```
## Residuals:
##      Min       1Q  Median       3Q      Max
## -28.291  -6.952   0.453   5.785  23.980
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.03188   17.60076   0.286    0.776
## feed         0.24323    0.02594   9.378 1.98e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.19 on 48 degrees of freedom
## Multiple R-squared:  0.6469, Adjusted R-squared:  0.6396
## F-statistic: 87.95 on 1 and 48 DF,  p-value: 1.985e-12
```

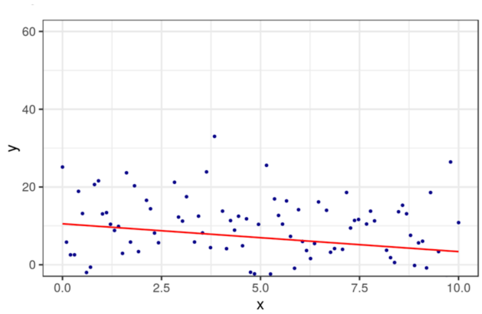
***Step 4** แปลความผลจากตาราง summary*



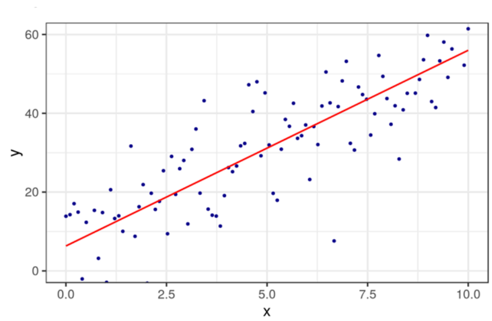
slope = 5.01 R^2 = 0.99 P < 2.2 x 10^{-16}



slope = 0.74 R^2 = 0.38 P < 4.75 x 10^{-12}



slope = 0.16 R^2 = -0.008 P = 0.66



slope = 4.73 R^2 = 0.65 P < 2.2 x 10^{-16}

Slope:	ถ้าเป็น +	ถ้าเป็น –
	ถ้าเป็นหรือใกล้ 0	
R-square:	ใกล้ 0	ใกล้ 1
P-value slope:	P ≤ 0.05	
	P > 0.05	