

## 1. เรียกข้อมูลจากไฟล์ **students.csv**

จงแสดงกราฟ (เลือกประเภทกราฟตามความเหมาะสม)

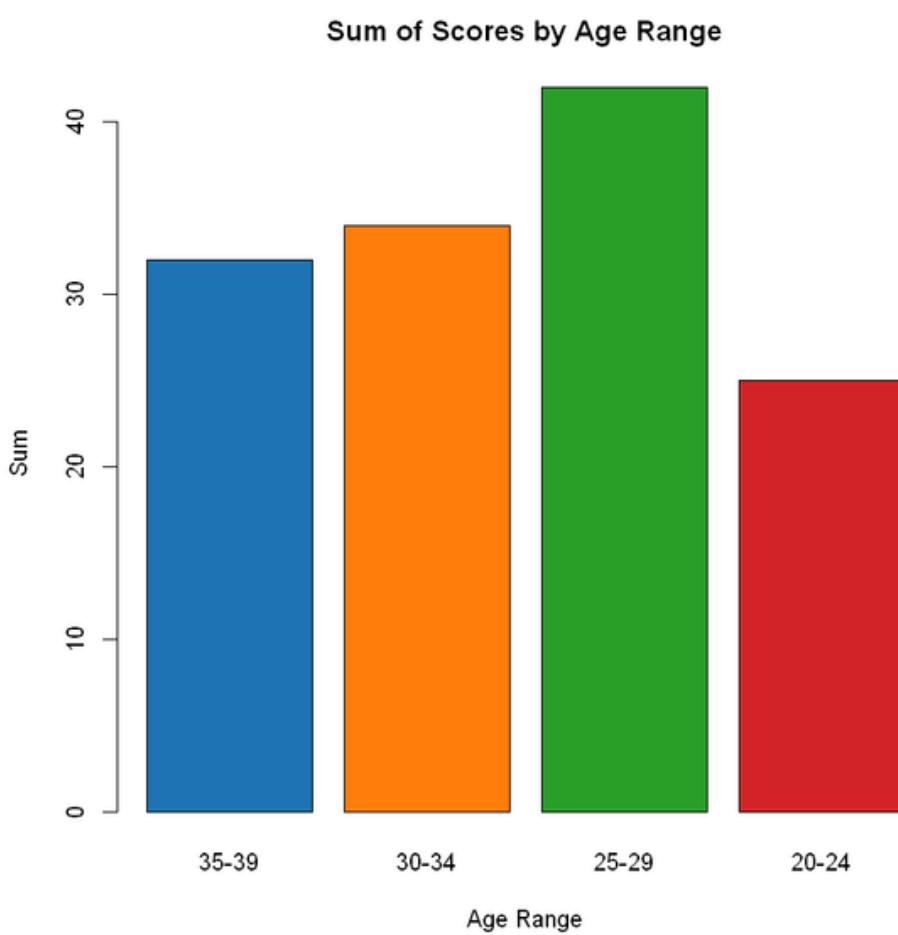
- เปรียบเทียบ จำนวน นศ ตามช่วงอายุ

```
# 1.1
# โหลดข้อมูลจากไฟล์ CSV ชื่อ "students.csv" โดยให้ตัวแปร students เท่ากับข้อมูลที่โหลดมา
students <- read.csv("students.csv")

# คำนวณผลรวมของคอลัมน์ 'O', 'A', 'B', 'AB' สำหรับแต่ละแกร์ และเก็บผลลัพธ์ในคอลัมน์ใหม่ชื่อ 'sum'
students$sum <- rowSums(students[, c('O', 'A', 'B', 'AB')])

# กำหนดสีที่แตกต่างกันสำหรับแท่งกราฟ --> สีน้ำเงิน, สีส้ม, สีเขียว, สีแดง
contrast_colors <- c("#1f77b4", "#ff7f0e", "#2ca02c", "#d62728")

# สร้างแผนภูมิแท่งของผลรวมโดยใช้ช่วงอายุเป็น Label บนแกน X
barplot(students$sum, # ข้อมูลที่จะแสดงในแผนภูมิแท่ง
        names.arg = students$Age, # ใช้คอลัมน์ Age เป็น x Label
        main = "Sum of Scores by Age Range", # ชื่อแผนภูมิ
        xlab = "Age Range", # x Label
        ylab = "Sum", # y Label
        col = contrast_colors[1:length(students$sum)]) # ใช้สีที่กำหนดไว้กับแท่งกราฟ
```



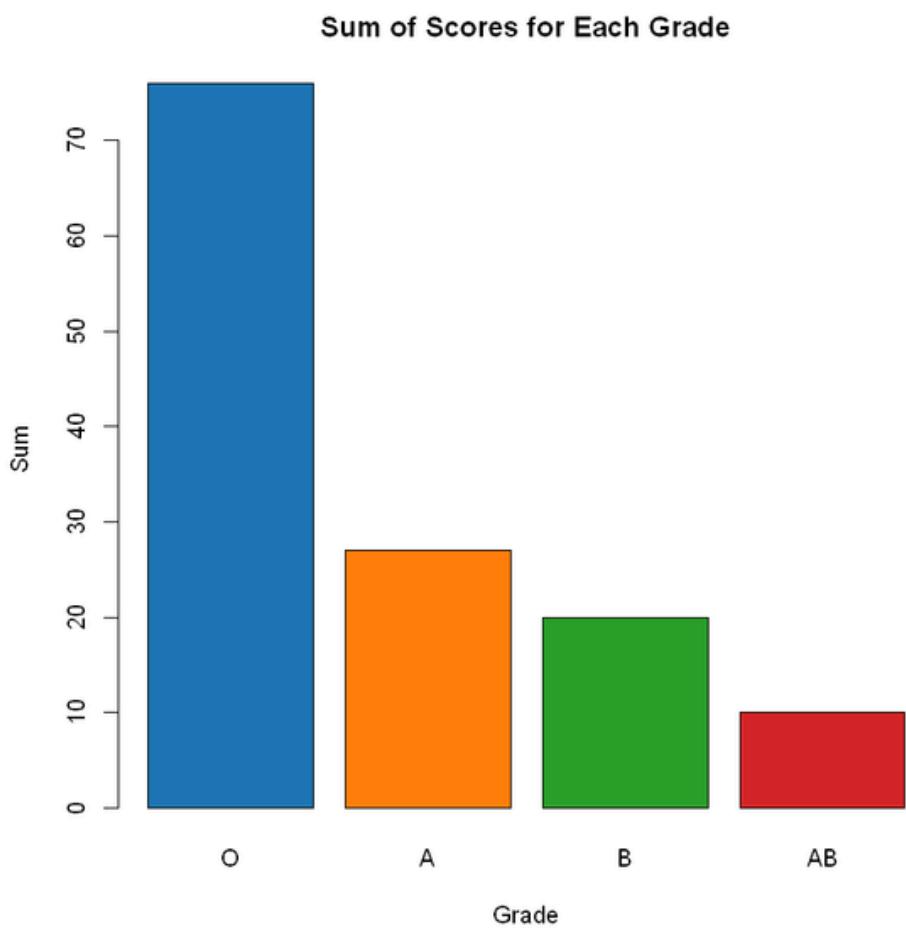
- เปรียบเทียบ กรุ๊ปเลือด ว่า แต่ละกรุ๊ปมีจำนวนกี่คน

```
# 1.2
# โหลดข้อมูลจากไฟล์ CSV ชื่อ "students.csv" โดยให้ตัวแปร students เท่ากับข้อมูลที่โหลดมา
students <- read.csv("students.csv")

# คำนวณผลรวมสำหรับแต่ละคอลัมน์ 'O', 'A', 'B', 'AB' และเก็บผลลัพธ์ในตัวแปร sumG
sumG <- colSums(students[c('O', 'A', 'B', 'AB')])

# กำหนดสีที่แตกต่างกันสำหรับแท่งกราฟ --> สีน้ำเงิน, สีเหลือง, สีเขียว, สีแดง
contrast_colors <- c("#1f77b4", "#ff7f0e", "#2ca02c", "#d62728")

# สร้างแผนภูมิแท่งของผลรวม โดยใช้กรุ๊ปเลือดเป็น Label บนแกน x
barplot(sumG, # ข้อมูลที่จะแสดงในแผนภูมิแท่ง
        names.arg = c('O', 'A', 'B', 'AB'), # ป้ายกำกับแกน x
        main = "Sum of Scores for Each Grade", # ชื่อแผนภูมิ
        xlab = "Grade", # x Label
        ylab = "Sum", # y Label
        col = contrast_colors) # ใช้สีที่กำหนดไว้กับแท่งกราฟ
```



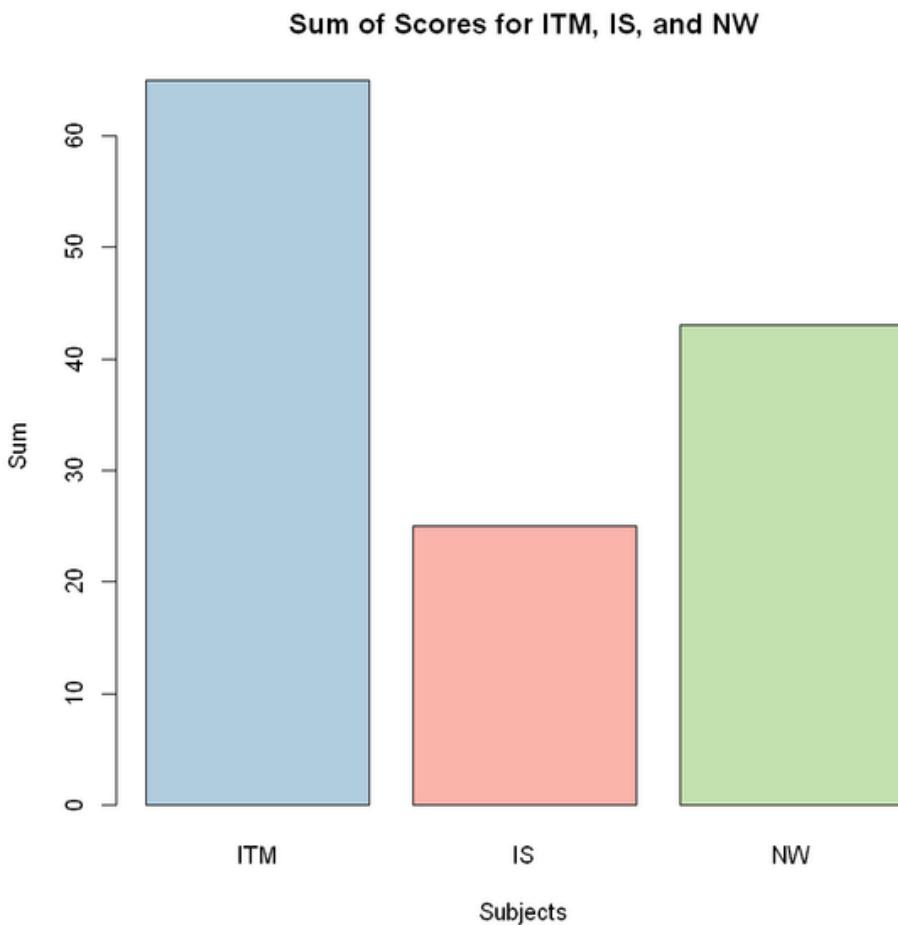
- เปรียบเทียบ Major ของ นศ ว่า เรียน major ละกี่คน

```
#· 1.3
#· โหลดข้อมูลจากไฟล์ CSV ชื่อ "students.csv" โดยให้ตัวแปร students เท่ากับข้อมูลที่โหลดมา
students <- read.csv("students.csv")

#· คำนวณผลรวมสำหรับแต่ละคอลัมน์ 'ITM', 'IS', 'NW' และเก็บผลลัพธ์ในตัวแปร sumM
sumM <- colSums(students[c('ITM', 'IS', 'NW')])

#· กำหนดสีพาสเทลสำหรับแท่งกราฟ--> สีฟ้าอ่อน, สีชมพูอ่อน, สีเขียวอ่อน
pastel_colors <- c("#B3CDE0", "#FBB4AE", "#C5E2B3")

#· สร้างแผนภูมิแท่งของผลรวมจำนวน Major ITM, IS, และ NW
barplot(sumM, #· ข้อมูลที่จะแสดงในแผนภูมิแท่ง
        names.arg = c('ITM', 'IS', 'NW'), #· x·Label
        main = "Sum of Scores for ITM, IS, and NW", #· ชื่อแผนภูมิ
        xlab = "Subjects", #· x·Label
        ylab = "Sum", #· y·Label
        col = pastel_colors) #· ใช้สีที่กำหนดไว้กับแท่งกราฟ
```



## 2. เรียกข้อมูลจากไฟล์ quality.csv

ชึ้งเก็บข้อมูล เสียง รบกวน และ มลพิษ

ใน 9 เมือง

จังคำนวนหา

-ค่าเฉลี่ยของมลพิษในแต่ละประเภท และ พล็อตกราฟ (เลือกประเภทกราฟตามความเหมาะสม)

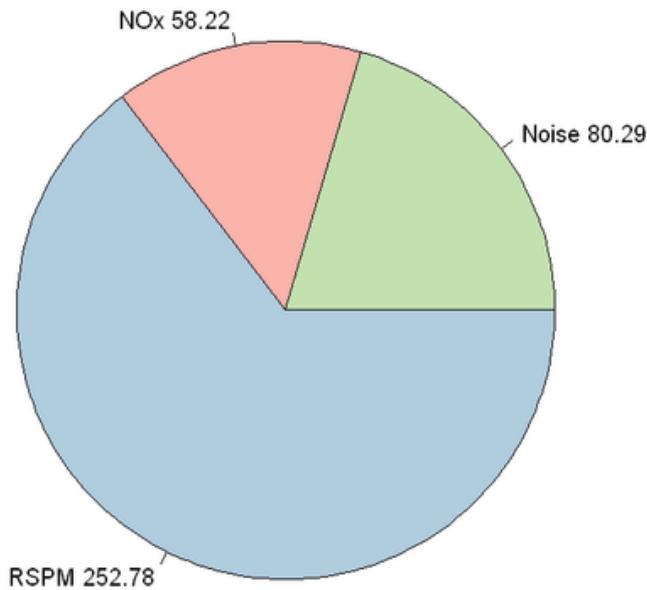
```
#· 2.1
#· โหลดข้อมูลจากไฟล์ CSV· ชื่อ· "quality.csv"· โดยให้ตัวแปร· quality· เท่ากับข้อมูลที่โหลดมา
quality <- read.csv("quality.csv")

#· คำนวณค่าเฉลี่ยระดับมลพิษสำหรับแต่ละพารามิเตอร์ (Noise, · NOx, · RSPM) · และเก็บผลลัพธ์ในตัวแปร· pollution
pollution <- colMeans(quality[c('Noise', 'NOx', 'RSPM')])

#· กำหนดสีพาสเทลสำหรับแผนภูมิวงกลม· --> · สีเขียวอ่อน, · สีชมพูอ่อน, · สีฟ้าอ่อน
pastel_colors <- c("#C5E2B3", "#FBB4AE", "#B3CDE0")

#· สร้างแผนภูมิวงกลมของค่าเฉลี่ยระดับมลพิษ· โดยใช้ชื่อพารามิเตอร์เป็น· Label
pie(pollution, · #· ข้อมูลที่จะแสดงในแผนภูมิวงกลม
    · · labels = paste(names(pollution), round(pollution, 2)), · #· Label· และค่าของแต่ละพารามิเตอร์
    · · main = "Average Pollution Levels", · #· ชื่อแผนภูมิ
    · · col = pastel_colors) · #· ໃบสีที่กำหนดไว้กับแผนภูมิวงกลม
```

Average Pollution Levels

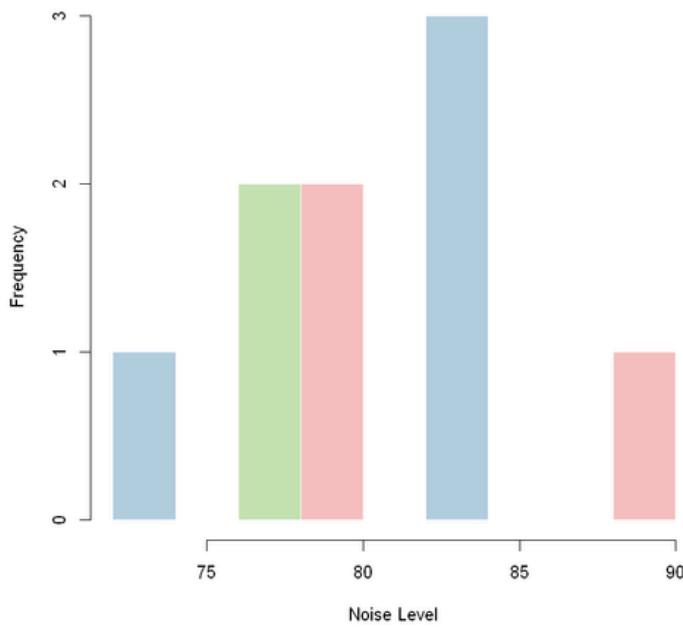


## -สร้าง histogram ของข้อมูลเสียงรบกวน

```
# 2.2
# โหลดข้อมูลจากไฟล์ CSV ชื่อ "quality.csv" โดยให้ตัวแปร quality เท่ากับข้อมูลที่โหลดมา
quality <- read.csv("quality.csv")

# สร้าง histogram ของระดับเสียงรบกวน
hist(quality$Noise, # ข้อมูลระดับเสียงรบกวนที่จะแสดงใน histogram
     main = "Distribution of Noise Levels", # ชื่อแผนภูมิ
     xlab = "Noise Level", # x Label
     ylab = "Frequency", # y Label
     col = c("#B3CDE0", "#FBB4AE", "#C5E2B3", "#F6C0C0", "#E2B2D2"), # ใช้สีที่กำหนดไว้
     border = "white", # สีขอบของแท่ง histogram
     breaks = 10) # จำนวนช่วงข้อมูล
```

Distribution of Noise Levels

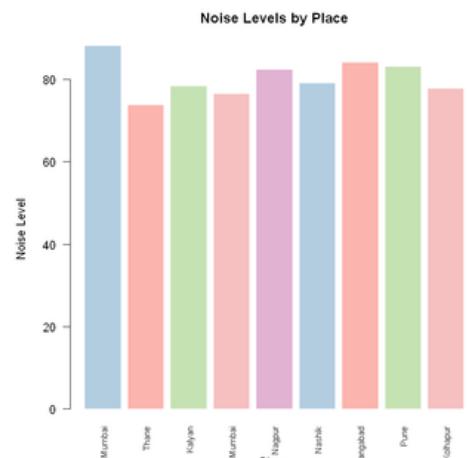


จำนวนช่วงข้อมูลอยู่ที่ 10

## \*ลองใช้ Barplot

```
# 2.2
# โหลดข้อมูลจากไฟล์ CSV ชื่อ "quality.csv" โดยให้ตัวแปร quality เท่ากับข้อมูลที่โหลดมา
quality <- read.csv("quality.csv")

# สร้างแผนภูมิแท่งของระดับเสียงรบกวน โดยใช้ชื่อสถานที่เป็น Label บนแกน x
barplot(quality$Noise, # ข้อมูลระดับเสียงรบกวนที่จะแสดงในแผนภูมิแท่ง
        names.arg = quality$Place, # ไม่ค่อสั้นชื่อ Place เป็น x Label
        main = "Noise Levels by Place", # ชื่อแผนภูมิ
        xlab = "Place", # x Label
        ylab = "Noise Level", # y Label
        las = 2, # หมุน Label บน x 90 องศา
        cex.names = 0.75, # ลดขนาดตัวอักษรของ Label บน x
        col = c("#B3CDE0", "#FBB4AE", "#C5E2B3", "#F6C0C0", "#E2B2D2"), # ใช้สีที่กำหนดไว้กับแท่งกราฟ
        border = NA) # ลบเส้นขอบของแท่งกราฟเพื่อให้ดูสะอาดตาขึ้น (ผนวลาสะดวกซึ่งนิคเน็มครับ)
```



## – ระดับเสียงรบกวน และมลพิษ เทียบกับของแต่ละเมือง และ พล็อตกราฟ (เลือกประเภทกราฟ)

```
#· 2.3
#· โหลดข้อมูลจากไฟล์ CSV ชื่อ "quality.csv" โดยให้ตัวแปร quality เท่ากับข้อมูลที่โหลดมา
quality <- read.csv("quality.csv")

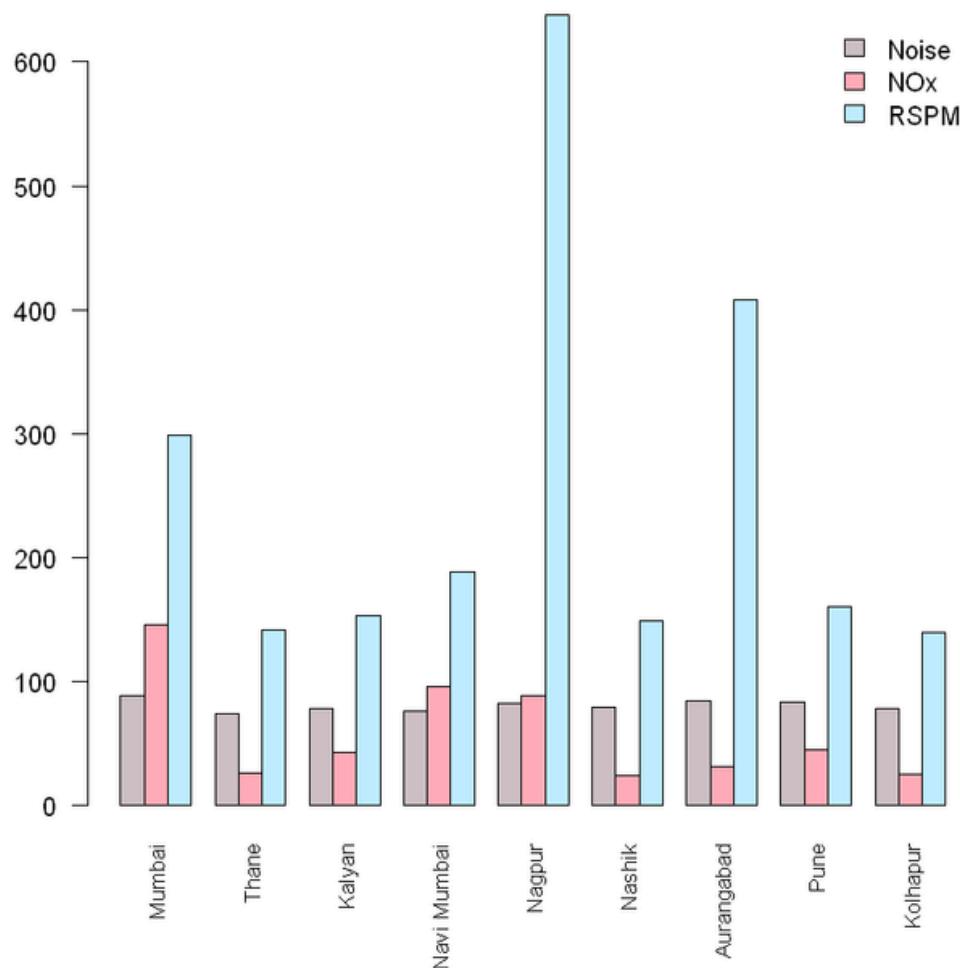
#· ตัดคอลัมน์ที่เกี่ยวกับไวรัส (คอลัมน์ที่ 2-ถึง 4) และเก็บในตัวแปร ql
feature <- quality[c(2:4)]

#· กำหนดชื่อแກของ data.frame ให้เป็นค่าจากคอลัมน์ 'Place'
row.names(feature) <- quality[["Place"]]

#· สลับແກ່และคอลัมน์ของข้อมูลเพื่อใช้ในการสร้างแผนภูมิแท่ง
feature <- t(feature)

#· สร้างแผนภูมิแท่งของระดับเสียงรบกวน, NOx, และ RSPM ตามสถานที่
bar <- barplot(feature, #· ข้อมูลที่จะแสดงในแผนภูมิแท่ง
               beside = TRUE, #· แสดงแท่งกราฟแยกกันสำหรับแต่ละกลุ่ม
               main = "Noise Level, NOx, and RSPM by Place", #· ชื่อแผนภูมิ
               col = c("lavenderblush3", "lightpink1", "lightblue1"), #· สีของแท่งกราฟ
               legend.text = c("Noise", "NOx", "RSPM"), #· ข้อความในค่าอธิบายสัญลักษณ์
               args.legend = list(x = "topright", bty = "n"), #· ตำแหน่งและรูปแบบของค่าอธิบายสัญลักษณ์
               las = 2, #· หมุน label แกน x 90 องศา
               cex.names = 0.85) #· ปรับขนาดตัวอักษรของ label แกน x
```

Noise Level, NOx, and RSPM by Place



# แบบฝึกหัด

ในโปรแกรม R จะมี Built in Data ชื่อ UScereal

```
> library('MASS')
> data(UScereal)
> attach(UScereal)
> names(UScereal) # to see the names
```

```
# 3.1
# โหลด Package MASS ซึ่งมีข้อมูลตัวอย่างหล้ายขุด รวมถึง UScereal
library(MASS)

# โหลดขุดข้อมูล UScereal จาก Package MASS
data(UScereal)

# ผูกขุดข้อมูล UScereal เข้ากับ R.session นั้นๆ --> ทำให้สามารถเข้าถึงตัวแปรในขุดข้อมูลได้โดยตรง โดยไม่ต้องใช้เครื่องหมาย $ 
attach(UScereal)

# แสดงชื่อคอลัมน์ทั้งหมดในขุดข้อมูล UScereal
names(UScereal)
# 0.0s
'mfr' 'calories' 'protein' 'fat' 'sodium' 'fibre' 'carbo' 'sugars' 'shelf' 'potassium' 'vitamins'
```

-Plot กราฟการกระจายข้อมูล จงหา

- 1) The relationship between manufacturer and shelf**
- 2) The relationship between fat and vitamins**
- 3) the relationship between fat and shelf**
- 4) the relationship between carbohydrates and sugars**
- 5) the relationship between fibre and manufacturer**
- 6) the relationship between sodium and sugars**

- 1) # สร้างกราฟการกระจายข้อมูลแสดงความลับพันธุระหว่างผู้ผลิต (manufacturer) และชั้นวางสินค้า (shelf)
 

```
plot(mfr, shelf, xlab = "Manufacturer", ylab = "Shelf", main = "The relationship between manufacturer and shelf")
```
- 2) # สร้างกราฟการกระจายข้อมูลแสดงความลับพันธุระหว่างไขมัน (fat) และวิตามิน (vitamins)
 

```
plot(fat, vitamins, xlab = "Fat", ylab = "Vitamins", main = "The relationship between fat and vitamins")
```
- 3) # สร้างกราฟการกระจายข้อมูลแสดงความลับพันธุระหว่างไขมัน (fat) และชั้นวางสินค้า (shelf)
 

```
plot(fat, shelf, xlab = "Fat", ylab = "Shelf", main = "The relationship between fat and shelf")
```
- 4) # สร้างกราฟการกระจายข้อมูลแสดงความลับพันธุระหว่างคาร์โบไฮเดรต (carbohydrates) และน้ำตาล (sugars)
 

```
plot(carbo, sugars, xlab = "Carbohydrates", ylab = "Sugars", main = "The relationship between carbohydrates and sugars")
```
- 5) # สร้างกราฟการกระจายข้อมูลแสดงความลับพันธุระหว่างใยอาหาร (fibre) และผู้ผลิต (manufacturer)
 

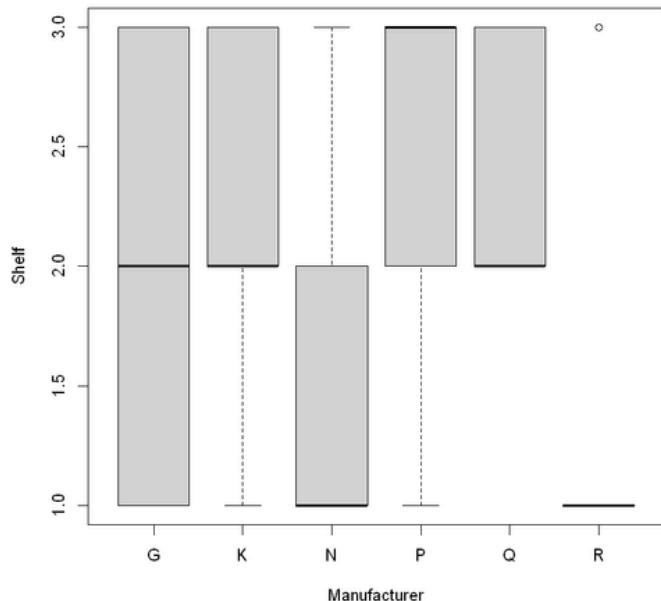
```
plot(fibre, mfr, xlab = "Fibre", ylab = "Manufacturer", main = "The relationship between fibre and manufacturer")
```
- 6) # สร้างกราฟการกระจายข้อมูลแสดงความลับพันธุระหว่างโซเดียม (sodium) และน้ำตาล (sugars)
 

```
plot(sodium, sugars, xlab = "Sodium", ylab = "Sugars", main = "The relationship between sodium and sugars")
```

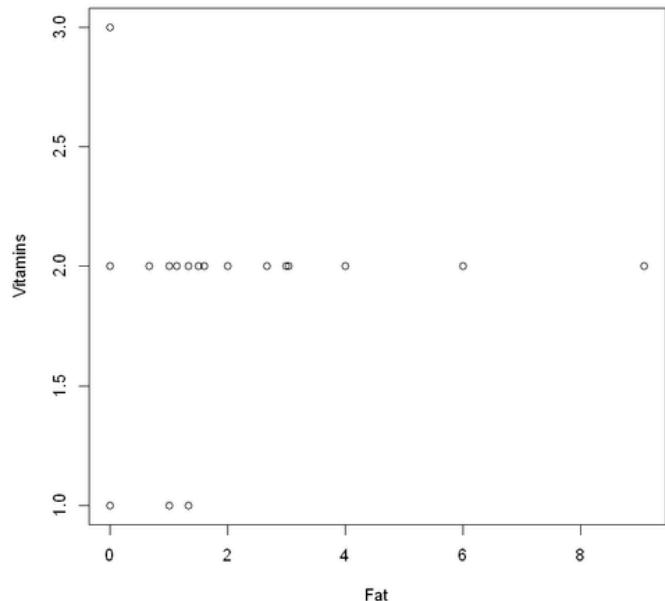
```
# ยกเลิกการผูกขุดข้อมูล UScereal
detach(UScereal)
```

**1)**

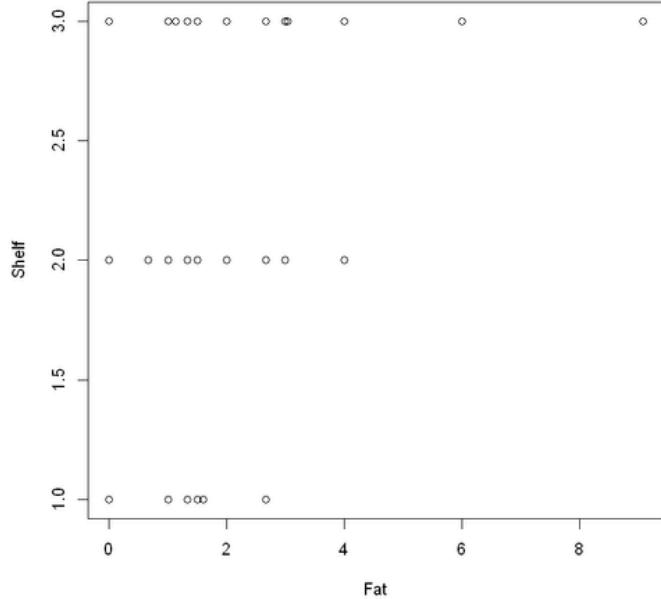
The relationship between manufacturer and shelf

**2)**

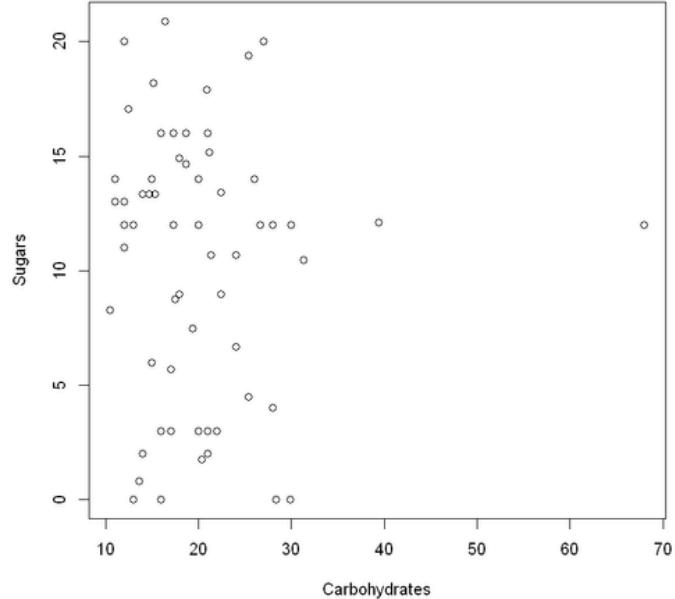
The relationship between fat and vitamins

**3)**

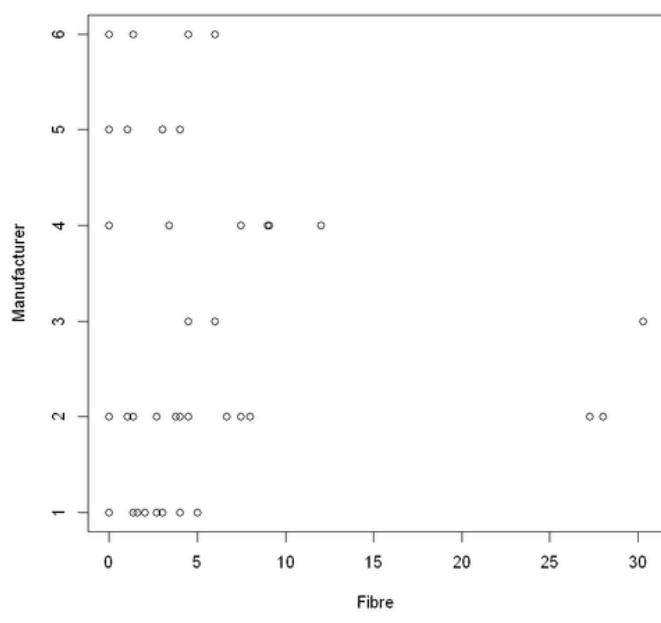
The relationship between fat and shelf

**4)**

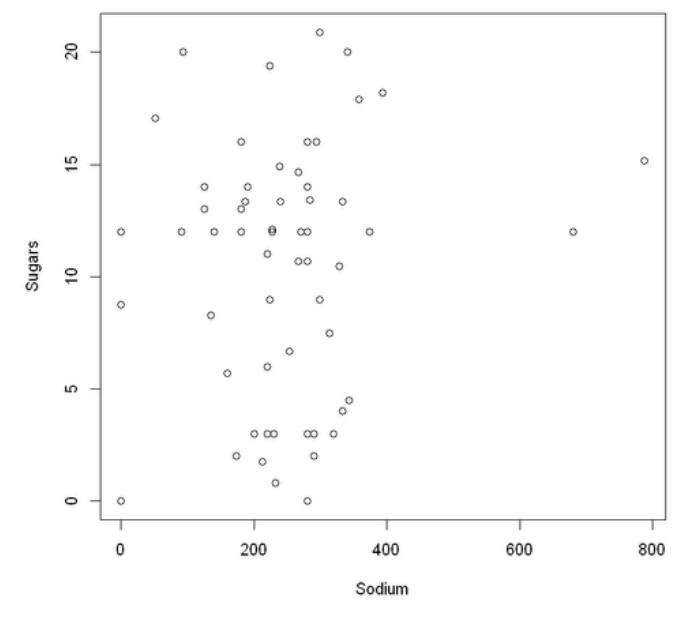
The relationship between carbohydrates and sugars

**5)**

The relationship between fibre and manufacturer

**6)**

The relationship between sodium and sugars



# แบบฝึกหัด

ในโปรแกรม R จะมี Built in Data ชื่อ mammals ที่เก็บข้อมูล body weight และ brain weight

-Plot กราฟการกระจายข้อมูล

```
#· 3.2
#· โหลด·Package·MASS·ซึ่งมีบุคคลข้อมูลตัวอย่างหลายบุคคล·รวมถึง·mammals
library('MASS')

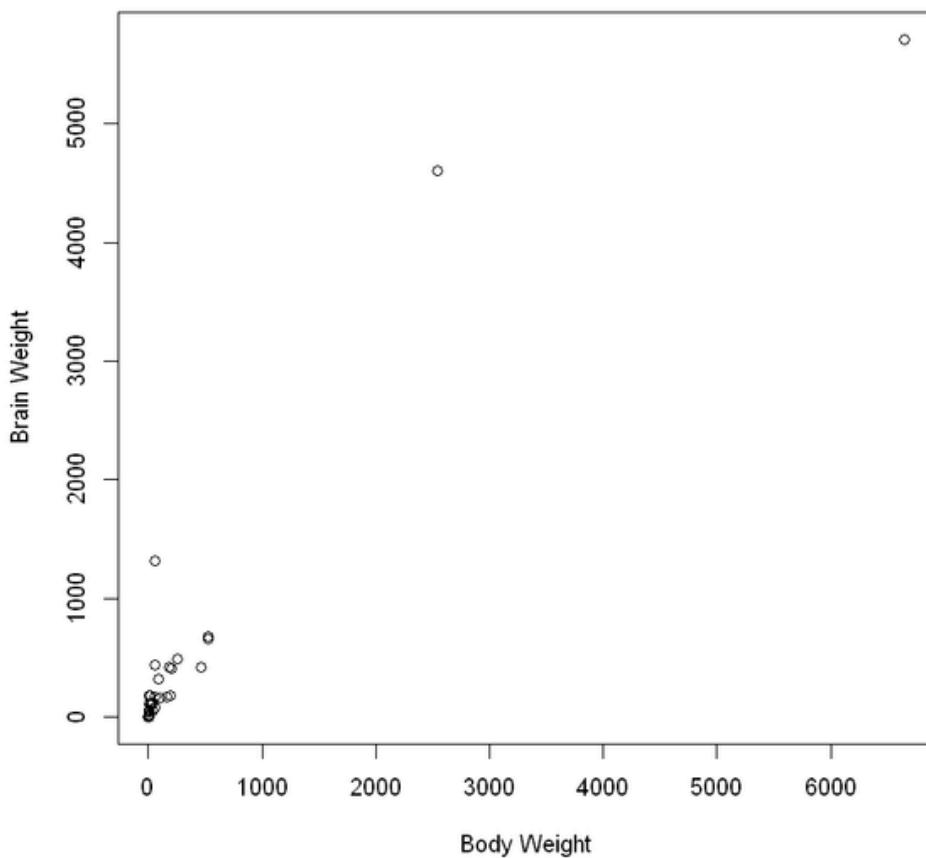
#· โหลดบุคคลข้อมูล·mammals·จาก·Package·MASS
data(mammals)

#· ผูกบุคคลข้อมูล·mammals·เข้ากับ·R·session·บัญชีนั้น-->·ทำให้สามารถเข้าถึงตัวแปรในบุคคลข้อมูลได้โดยตรง·โดยไม่ต้องใช้เครื่องหมาย·$·
attach(mammals)

#· แสดงชื่อคอลัมน์ทั้งหมดในบุคคลข้อมูล·mammals
names(mammals)
✓ 0.0s
'body' · 'brain'
```

```
#· สร้างกราฟการกระจายข้อมูลแสดงความสัมพันธ์ระหว่างน้ำหนักตัวและน้ำหนักสมอง
plot(body, brain, #· ข้อมูลที่จะแสดงในกราฟ
      xlab = "Body Weight", #· x·Label
      ylab = "Brain Weight", #· y·Label
      main = "The relationship between body weight and brain weight") #· ชื่อกราฟ
```

The relationship between body weight and brain weight



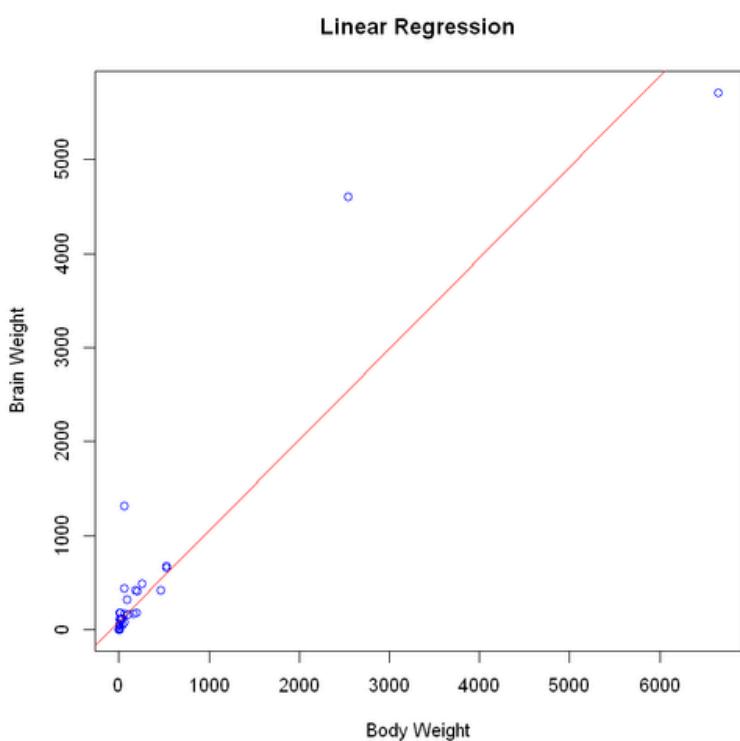
## -หัวสมการ Linear Regression และ plot กราฟ

```
#· สร้างโมเดลการ回帰เชิงเส้น (Linear Regression)
lm_model <- lm(brain ~ body, data = mammals)

#· แสดงสมการการ回帰เชิงเส้น
cat("Linear Regression Equation: brain =", round(lm_model$coefficients[1], 2), "+", round(lm_model$coefficients[2], 2), "* body\n")

#· สร้างกราฟการกระจายและเส้นการ回帰
plot(body, brain, xlab = "Body Weight", ylab = "Brain Weight", main = "Linear Regression", col = "blue")
abline(lm_model, col = "red")
```

Linear Regression Equation: brain = 91 + 0.97 \* body



## -หาค่าสัมประสิทธิ์ Correlation

```
#· คำนวณค่าสัมประสิทธิ์สหสัมพันธ์ระหว่างน้ำหนักตัวและน้ำหนักสมอง
correlation_coefficient <- cor(mammals$body, mammals$brain)
```

```
#· แสดงค่าสัมประสิทธิ์สหสัมพันธ์
correlation_coefficient
```

✓ 0.0s

0.934163842323355



ทดลองพื้นที่ต่อค่า  $\log$  (ข่ายปรับ scale ข้อมูล) ของแต่ละตัวแปร และ หาสมการ Linear Regression และ plot กราฟเปรียบเทียบ

```
#· สร้างกราฟการกระจายข้อมูลของค่า · Log · ของน้ำหนักตัวและน้ำหนักสมอง
plot(log(body), log(brain), #· ข้อมูลที่จะแสดงในกราฟ
     xlab = "Log Body Weight", #· x · Label
     ylab = "Log Brain Weight", #· y · Label
     main = "Log-Transformed Body and Brain Weight")
```



```
#· สร้างโมเดลการลด削除เชิงเส้นสำหรับข้อมูลที่แปลงด้วย · Log
lm_model_log <- lm(log(brain) ~ log(body), data = mammals)

#· สร้างกราฟการกระจายของข้อมูลที่แปลงด้วย · Log
plot(log(body), log(brain), #· ข้อมูลที่จะแสดงในกราฟ
     xlab = "Log Body Weight", #· x · Label
     ylab = "Log Brain Weight", #· y · Label
     main = "Log-Transformed Body and Brain Weight") #· ชื่อกราฟ

#· เพิ่มเส้นการลด削除ลงในกราฟ
abline(lm_model_log, col = "red")

#· และแสดงสมการการลด削除เชิงเส้น
cat("Linear Regression Equation (Log-Transformed): log(brain) = ",
    round(lm_model_log$coefficients[1], 2), "+",
    round(lm_model_log$coefficients[2], 2), "* log(body)\n")
```

Linear Regression Equation (Log-Transformed):  $\log(\text{brain}) = 2.13 + 0.75 * \log(\text{body})$

