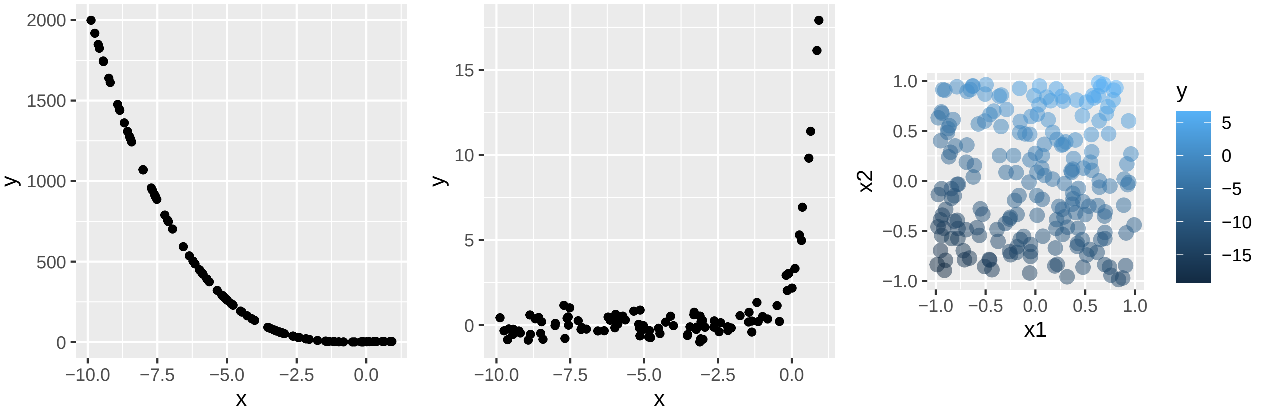
Course 2 Section 3.2 - STATISTICAL MODELS

Jiaying Wu

14/10/2020

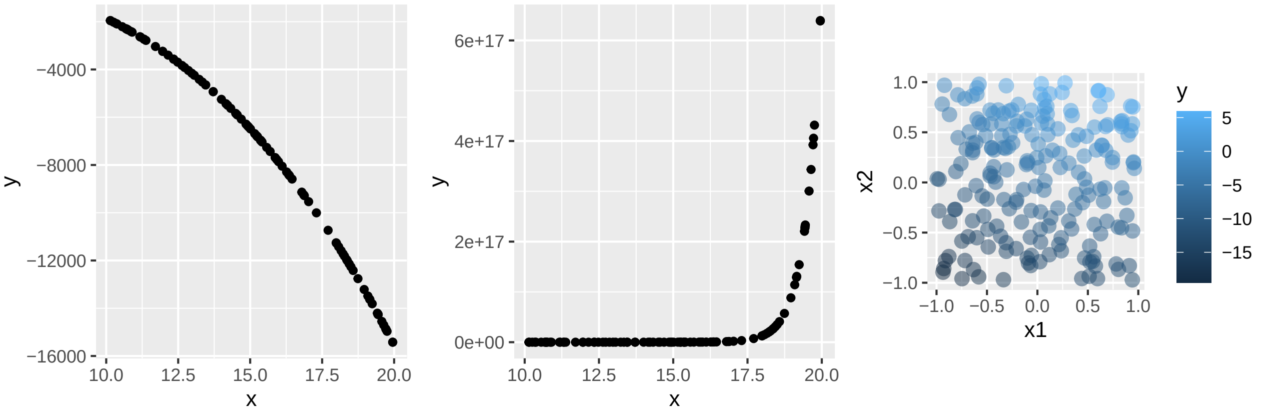
Q1: try the sample code to generate data with different functional forms

```r
# Load tidyverse
library(tidyverse)

# Load gridExtra
library(gridExtra)

# Randomly generate 100 values between -10 to 1 and store in x
x <- runif(100, -10, 1)

# Data frame of x and y (y based on a quadratic and cubic function of x)
# rnorm() adds noises to y (run rnorm(100)*0.2 to see what this noise looks like)
df <- tibble(x, y=2+3*x+x^2-2*x^3+rnorm(100)*0.2)

# Scatter plot of y against x
p1 <- ggplot(df, aes(x=x, y=y)) + geom_point()

# Data frame of x and y (y based on a exponential function of x)
df2 <- tibble(x, y=3*exp(2*x)+rnorm(100)*0.5)

# Scatter plot of y against x
p2 <- ggplot(df2, aes(x=x, y=y)) + geom_point()

# Randomly generate 200 values between -1 to 1 and store in x1
x1 <- runif(200, -1, 1)

# Randomly generate 200 values between -1 to 1 and store in x2
x2 <- runif(200, -1, 1)

# Data frame of x1, x2 and y (y based on x1 and x2)
df3 <- tibble(x1, x2, y=-5+4*x1-2*x1^2+10*x2+rnorm(200)*0.1)

# Scatter plot of x2 against x1 coloured by y
p3 <- ggplot(df3, aes(x=x1, y=x2, colour=y)) + geom_point(size=3, alpha=0.5) + theme(aspect.ratio=1)

# Place p1, p2 and p3 in a 3 column grid (grid.arrange() comes from gridExtra package)
grid.arrange(p1, p2, p3, ncol=3)
```



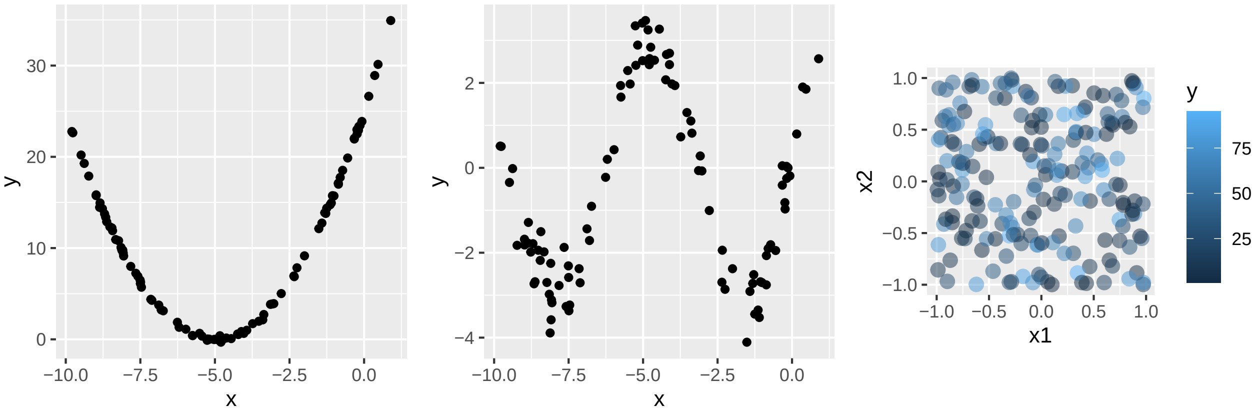
Q2.change the domain of the explanatory variable, and discover what happens to the shape of the data

```
# Randomly generate 100 values between -10 to 1 and store in x
x <- runif(100, 10, 20)

# Data frame of x and y (y based on a quadratic and cubic function of x)
# rnorm() adds noises to y (run rnorm(100)*0.2 to see what this noise looks like)
df <- tibble(x, y=2+3*x+x^2-2*x^3+rnorm(100)*0.2)

# Scatter plot of y against x
p1 <- ggplot(df, aes(x=x, y=y)) + geom_point()

# Data frame of x and y (y based on a exponential function of x)
df2 <- tibble(x, y=3*exp(2*x)+rnorm(100)*0.5)

# Scatter plot of y against x
p2 <- ggplot(df2, aes(x=x, y=y)) + geom_point()

# Randomly generate 200 values between -1 to 1 and store in x1
x1 <- runif(200, -1, 1)

# Randomly generate 200 values between -1 to 1 and store in x2
x2 <- runif(200, -1, 1)

# Data frame of x1, x2 and y (y based on x1 and x2)
df3 <- tibble(x1, x2, y=-5+4*x1-2*x1^2+10*x2+rnorm(200)*0.1)

# Scatter plot of x2 against x1 coloured by y
p3 <- ggplot(df3, aes(x=x1, y=x2, colour=y)) + geom_point(size=3, alpha=0.5) + theme(aspect.ratio=1)

# Place p1, p2 and p3 in a 3 column grid (grid.arrange() comes from gridExtra package)
grid.arrange(p1, p2, p3, ncol=3)
```



Q3.challenge yourself - create some new functions and simulate some samples.

```
# Randomly generate 100 values between -10 to 1 and store in x
x <- runif(100, -10, 1)

# Data frame of x and y (y based on a quadratic and cubic function of x)
# rnorm() adds noises to y (run rnorm(100)*0.2 to see what this noise looks like)
df <- tibble(x, y = (x+5)^2+rnorm(100)*0.2)

# Scatter plot of y against x
p1 <- ggplot(df, aes(x=x, y=y)) + geom_point()

# Data frame of x and y (y based on a exponential function of x)
df2 <- tibble(x, y= 3*sin(x)+rnorm(100)*0.5)

# Scatter plot of y against x
p2 <- ggplot(df2, aes(x=x, y=y)) + geom_point()

# Randomly generate 200 values between -1 to 1 and store in x1
x1 <- runif(200, -1, 1)

# Randomly generate 200 values between -1 to 1 and store in x2
x2 <- runif(200, -1, 1)

# Data frame of x1, x2 and y (y based on x1 and x2)
df3 <- tibble(x1, x2, y=cos(x)+x^2+rnorm(200)*0.1)

# Scatter plot of x2 against x1 coloured by y
p3 <- ggplot(df3, aes(x=x1, y=x2, colour=y)) + geom_point(size=3, alpha=0.5) + theme(aspect.ratio=1)

# Place p1, p2 and p3 in a 3 column grid (grid.arrange() comes from gridExtra package)
grid.arrange(p1, p2, p3, ncol=3)
```


y
30
20
10
0
−10.0
−7.5
−5.0
−2.5
0.0
x
y
2
0
−2
−4
−10.0
−7.5
−5.0
−2.5
0.0
x
x2
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
x1
y
75
50
25