Normal Distribution Vignette

Oscar A. Trevizo

2023-05-09

Contents

1 Build a normally distributed dataset

Use y <- rnorm(100) to generate a random sample of size 100 from a normal distribution.

```
#R Code here

# Lets assign the sample size of 100 to variable "S" to reuse it in this assignment.
S <- 100

# Now get S number of Normally distributed random numbers.
y <- rnorm(S)
```

2 Calculate mean and sd

```
#R code here

# Mean and Std. Deviation funcitons:
mu <- mean(y)
sigma <- sd(y)

# Let's print them using the paste() function.
# Keep the display to 2 decimal points.
paste(' Mean of y = ', round(mu, digits=2))
```

```
## [1] " Mean of y =  -0.02"
```

```
paste(' Standard deviation of y = ', round(sigma, digits=2))
```

```
## [1] " Standard deviation of y =  0.91"
```

- The *rnorm(100)* function generates 100 Normally distributed random numbers with *mean* = 0 and standard deviation $sd = 1$.
- While the theoretical *mean* and *standard deviation* in *rnorm()* are 0 and 1 respectively, the experiments (each try or trial) do not result in those exact values due to randomness.
- Therefore, the resulting *mean* is close to 0, and the resulting *standard deviation* is close to 1, but not necessarily exactly 0 and 1 respectively due to randomness.
- The *paste()* function printed two results of the *mean* and *standard deviation*.
- Note: I will be using the notation *sigma* to refer to the standard deviation in much of my code.

Run it N times (make it 30 for this vignette). Store the N means in a vector. Verify the standard deviation of the values.

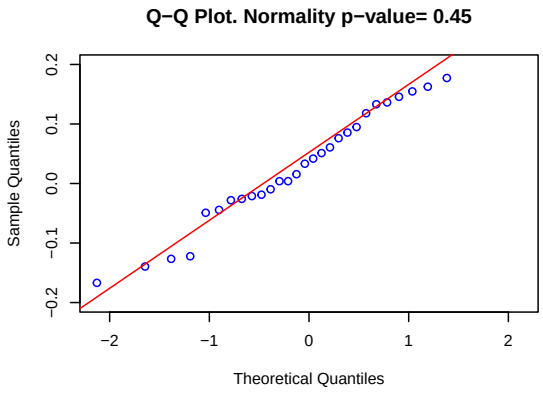
```
#R code here

# Let's assign our 30 experiments to a variable "N".
N <- 30

# Initialize vector MEAN of size 30
MEAN <- numeric(N)

# Run a loop that gets a new data sample, calculates the mean and sd, and stores the result in MEAN.
for (i in 1:N) {
  # MEAN is an indexed vector that will take values from A[1] to A[30] as we loop through.
  # Store the mean of a Normally distributed sample of size S into MEAN[i]. It will happen N times.
  MEAN[i] <- mean(rnorm(S))
}

# Coming out of the loop, we have a vector MEAN with 30 values.
# Calculate the standard deviation of MEAN.
SIGMA <- sd(MEAN)

# Display SD below
paste('The standard deviation of AGG is ', round(SIGMA, 2))
```

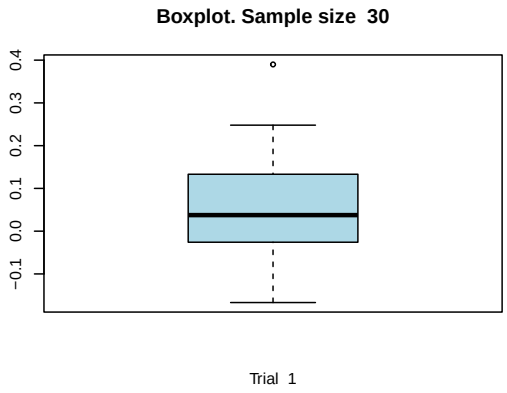
```
## [1] "The standard deviation of AGG is  0.09"
```

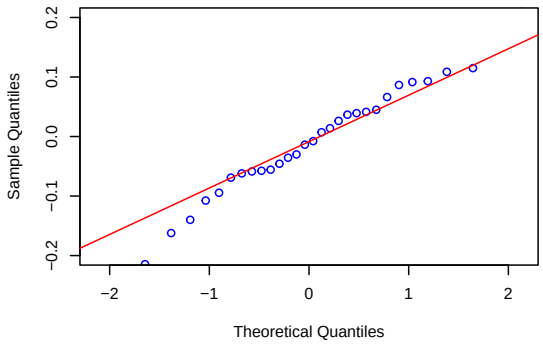
- *MEAN* is a vector with 30 values. Each value in the vector is the *mean* of 100 Normally distributed random numbers generated by *rnorm()*.
- *SIGMA* is a numeric variable that contains the *standard deviation* of those 30 values from vector *MEAN*.
- The result, the *standard deviation* of the *means* from those 30 *means* (the experiments or trials), is displayed above by the code.
- Notice how *SIGMA* is much smaller than the *standard deviation* obtained in the previous question where we were calculating the sigma of 100 Normally distributed random numbers.
- This time, *SIGMA* is calculated on the *means*, so it results in a smaller number, more narrow value.
- The reason *SIGMA* is so small is, the *means* tend to be close to 0, over and over. We are no longer measuring the *standard deviation* on the 100 random numbers, but on the *mean* of those numbers.

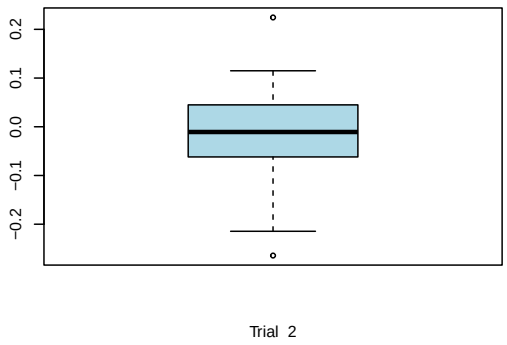
Run it multiple times (4 times in this vignette), showing each of the distributions of 30 means in a normal probability plot and box plot.

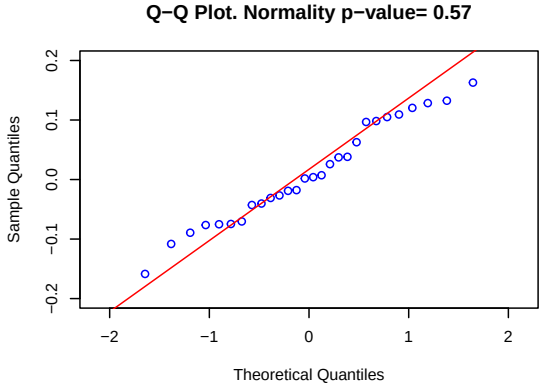
```
#R code here

par(mfrow = c(4, 2))

# Loop through the trial runs, or experiments.
for (i in 1:4) {
  # Run a loop that gets a new data sample, calculates the mean and sd, and stores the result in MEAN.
  # First get an N vector MEAN with the means of random numbers
  for (j in 1:N) {
    # MEAN is an indexed vector that will take values from A[1] to A[30] as we loop through.
    # Store the mean of a Normally distributed sample of size S into MEAN[i]. It will happen N times.
    MEAN[j] <- mean(rnorm(S))
  }
  SHAPIRO <- shapiro.test(MEAN)
  qqnorm(MEAN,
         ylim=c(-0.2, 0.2),
         col='blue',
         main=paste('Q-Q Plot. Normality p-value=', round(SHAPIRO$p.value, 2)))
  qqline(MEAN,
         ylim=c(-0.2, 0.2),
         col='red')
  boxplot(MEAN,
          col='lightblue',
          main=paste('Boxplot. Sample size ', N),
          xlab=paste('Trial ', i))
}
```

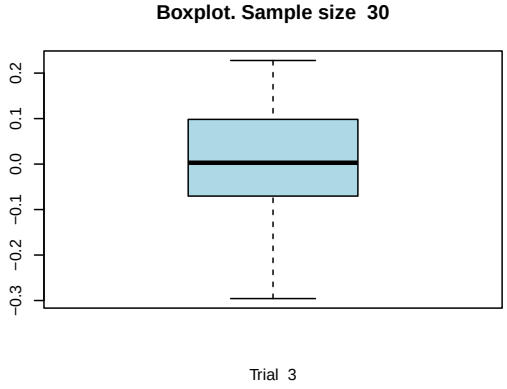

Q−Q Plot. Normality p−value= 0.45
Sample Quantiles
Theoretical Quantiles
−2
−1
0
1
2
−0.2
−0.1
0.0
0.1
0.2

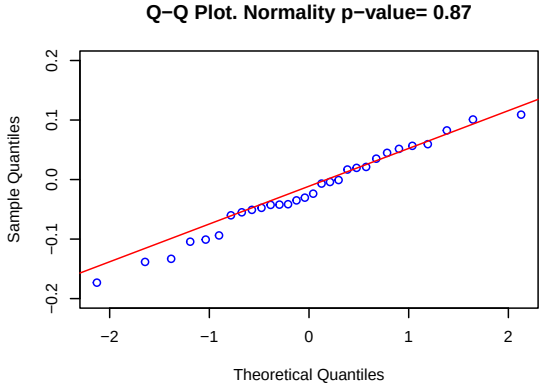

Boxplot. Sample size 30
Trial 1
−0.1
0.0
0.1
0.2
0.3
0.4

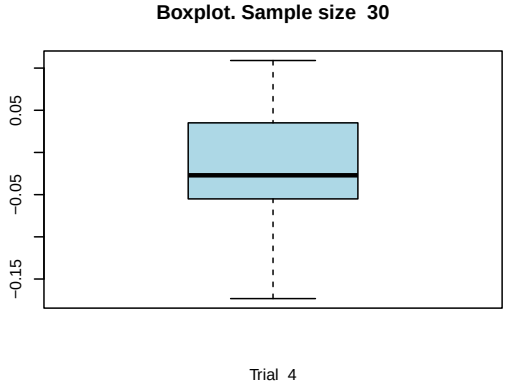

Q−Q Plot. Normality p−value= 0.88
Sample Quantiles
Theoretical Quantiles
−2
−1
0
1
2
−0.2
−0.1
0.0
0.1
0.2


Boxplot. Sample size 30
Trial 2
−0.2
−0.1
0.0
0.1
0.2


Q−Q Plot. Normality p−value= 0.57
Sample Quantiles
Theoretical Quantiles
−2
−1
0
1
2
−0.2
−0.1
0.0
0.1
0.2


Boxplot. Sample size 30
Trial 3
−0.3
−0.2
−0.1
0.0
0.1
0.2


Q−Q Plot. Normality p−value= 0.87
Sample Quantiles
Theoretical Quantiles
−2
−1
0
1
2
−0.2
−0.1
0.0
0.1
0.2


Boxplot. Sample size 30
Trial 4
−0.15
−0.05
0.05

Takeaways:

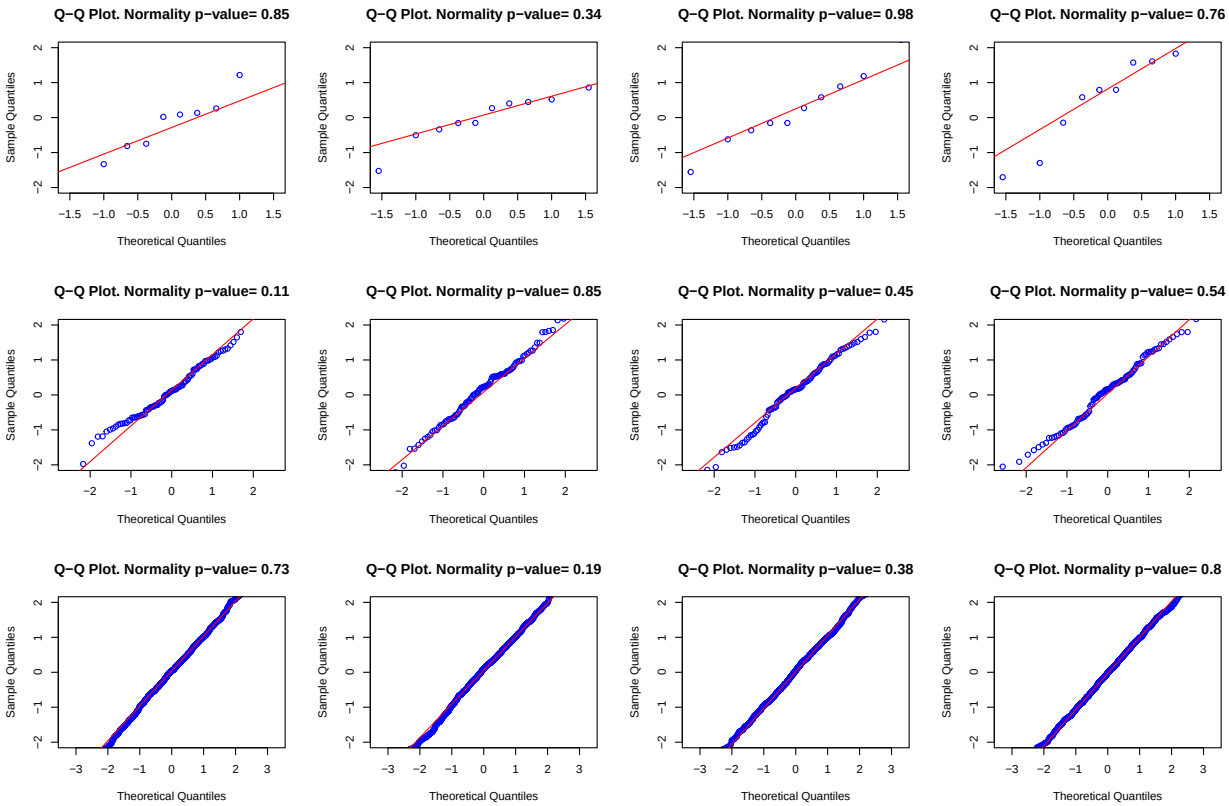
- The results provide 4 sets of 2 plots each: A *Normal Probability Plot (a.k.a. Q-Q Plot)* and a *boxplot* for each of the 4 experiments.
- I added a straight diagonal line to the *Q-Q plot* to help us visualize the results.
- In addition, I included *Shapiro-Wilk Tests* to test the H_o *null hypothesis* for each experiment run.
- The *null hypothesis* H_o of a *Shapiro-Wilk Test* states that the sample was generated from a *Normal distribution*. The H_A *alternate hypothesis* rejects the *null hypothesis* stating that the same did not come from a *Normal distribution*.
- We want to know if we should *reject* the *null hypothesis*, or keep the *null hypothesis*.
- The *null hypothesis* test has a key parameter called the *p-value*.
- If the *p-value* is smaller than 0.05 then we will reject the H_o *null hypothesis* and will propose that our sample does not come from a *Normal distribution*.
- And if the *p-value* is greater than 0.05, then we will keep the H_o *null_hypothesis*, stating that the *null hypothesis* is possible, and we propose the sample comes from a *Normally distribution*.
- If the distribution were to be Normal, the *Q-Q plot* would have the data points (*the dots*) aligned close to the straight diagonal line.
- We would also want to see *Shapiro-Wilk Test p-value* smaller than 0.05.
- **Results:**
- We do see a tendency of the dots following the *Q-Q line*, even though the dots are not exactly on top of the line in all four experiments.
- In fact, some experiments exhibit *tails* either at the start or at the end of the *Q-Q plot*. So we want to know more about the distribution.
- We will not get *all* of the points lined up directly on top pf the *Q-Q line* because there is always some noise or unexplained variation in the observations (hence being a *random* sample).
- The *Shapiro-Wilk Test p-values*, included on the title of each *Q-Q plot*, are all greater than 0.05. Therefore, we will keep the H_o *null hypothesis* and state that the samples came from a *Normal distribution*.
- The *boxplots* provide another visual to help us assess or form an opinion on whether the data may follow a *Normal* distribution or not.
- The *boxplots* are very consistent. The range between 1st and 3rd quartiles is *narrow* based on an observation: The *boxplots* have those quartiles between approximately -0.1 and 0.1 in all cases, sometimes even closer.
- Therefore, the *boxplots* support the notion that these sampes came from *Normal distributions*.
- In conclusion, these sample came from *Normal distributions*.

3 Calc. p-values with Shapiro-Wilk tests

3.1 Plot mltiple Q-Q plots multiple sample sizes

Show normal probability plots for multiple random samples (make it 4 here) of a size 10 for example.

```r
#R Code Here

par(mfrow = c(3, 4))

# S is a vector that contains the different sample sizes 10, 100, and 1000.
S <- c(10, 100, 1000)

# Now we loop through the 3 sets of 4 experiments each.
for (i in 1:3) {
  # Now run the 4 experiments
  for (j in 1:4) {
    # Run a trial sample size S[i] based on the S vector above.
    SAMPLE <- rnorm(S[i])
    SHAPIRO <- shapiro.test(SAMPLE)
    # ntest.p.value[i] <- SHAPIRO$p.value
    qqnorm(SAMPLE,
           ylim=c(-2, 2),
           col='blue',
           main=paste('Q-Q Plot. Normality p-value=', round(SHAPIRO$p.value, 2)))
    qqline(SAMPLE,
           ylim=c(-2, 2),
           col='red')
  }
}
```



- In all cases, *Shapiro-Wilk Tests* indicates that we should *keep* the *null Hypotheis* that our sample comes

from a *Normal distribution* and it does. But the degree in which it proposes such *null hypothesis*, based on the *p-value* varies as we increase the *sample size*.

- WE can see the impact from the *Central Limit Theorem (CLT)*. As we increase the *sample size* we get values that get closer to a *Normal distribution*.
- The top panel of 4 plots with *sample size* of 10 exhibits irregularity. We can see that in how the *data points* follow (or not follow) the *Q-Q line*. The *Q-Q line* is also irregular (notice the difference in slopes).
- The middle panel of 4 plots with *sample size* of 100 is an improvement over the first panel. But it still exhibits irregularities. We tend to have *tails* in either end of the plot.
- The bottom panel of 4 plots with *sample size* of 1000 offers a very evident *Normal distribution*. We can be highly confient that the samples came from a *Normal distribution*, without a question.