# 534hw4

2024-02-14

1. Simulation study. Generate a regression data set Yi = 2 * Xi + Wi of n = 100, where Wis are from Gamma distribution and Xis are from binomial distribution bin(10, p), then mimic Example 6 around pages 19-23, fitting normal distrition model and Gamma distribution model, respectively. Do qqplot, CB plots, MD plot, modified ks.test

```
set.seed(42)

n = 100
p = 0.5
alpha = 2
beta = 2

Xi = rbinom(n, size = 10, prob = p)

Wi = rgamma(n, shape = alpha, scale = beta)

Yi = 2 * Xi + Wi
```

```
model_normal = lm(Yi ~ Xi)
summary(model_normal)
```

```
## lm(formula = Yi ~ Xi)
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.3068     0.7930   5.431 4.08e-07 ***
## Xi            1.9686     0.1477  13.328  < 2e-16 ***
```

```
model_gamma = glm(Yi ~ Xi, family = "Gamma")
summary(model_gamma)
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.1280095  0.0054354   23.55   <2e-16 ***
## Xi          -0.0105476  0.0008881  -11.88   <2e-16 ***
```

---

**Teacher:** It is good to use glm to get the MLE, but we have not reached that section yet.

---

```
u = fitdistr(Yi,"normal")
qqPlot(Yi, distribution = "norm", mean = u$e[1], sd = u$e[2])
```

---

**Teacher:** It is better to use model_normal$residuals rather than Yi, then we do not need to bother with the effect of $X_i$'s.

---

```
## [1] 44 62
```

```
z = fitdistr(Yi, "gamma")
```

---

**Teacher:** Should use model_normal$residuals+ model_normal$co[1] rather than Yi, as $Y - 2X = W \sim$ Gamma(2,2).

---

```
qqPlot(Yi, distribution = "gamma", shape = z$e[1], scale =z$e[2])
```

```
## [1] 44 62
```

```
Yi = sort(Yi)
fitted_values = fitted(model_normal)
N = length(Yi)

plot(Yi, ppoints(Yi), type = "S", xlim = c(min(Yi), max(Yi)), lty = 3, xlab = "Yi (Observed)", ylab = "
s = 1.96 * sqrt(ppoints(Yi) * (1 - ppoints(Yi)) / N)
lines(Yi, ppoints(Yi) + s, type = "S", lty = 3)
lines(Yi, ppoints(Yi) - s, type = "S", lty = 3)

fitted_sorted = sort(fitted_values)
lines(fitted_sorted, ppoints(fitted_sorted), type = "S", lty = 1, col = "red")

leg.names = c("Yi", "Fitted Values")
legend("bottomright", legend = leg.names, col = c("black", "red"), lty = c(3, 1), cex = 1.0)
```

```
Yi = sort(Yi)
fitted_values = fitted(model_gamma)
N = length(Yi)

plot(Yi, ppoints(Yi), type = "S", xlim = c(min(Yi), max(Yi)), lty = 3, xlab = "Yi (Observed)", ylab = "
s = 1.96 * sqrt(ppoints(Yi) * (1 - ppoints(Yi)) / N)
lines(Yi, ppoints(Yi) + s, type = "S", lty = 3)
lines(Yi, ppoints(Yi) - s, type = "S", lty = 3)

fitted_sorted = sort(fitted_values)
lines(fitted_sorted, ppoints(fitted_sorted), type = "S", lty = 1, col = "red")

leg.names = c("Yi", "Fitted Values")
legend("bottomright", legend = leg.names, col = c("black", "red"), lty = c(3, 1), cex = 1.0)
```

```
x = model_normal$residuals
s = sd(x)
t = ks.test(x, "pnorm", 0, s)$p
n = length(Xi)
set.seed(42)
u = rep(0,1000)
for (i in 1:1000) {
  z = rnorm(n, mean = 0, sd = s)
  x = lm(z ~ 1)$resid
  v = sd(x)
  u[i] = ks.test(z, "pnorm", mean = 0, sd = v)$p.value
}
proportion = mean(u < t)
```

```
proportion
```

```
## [1] 0.268
```

---

**Teacher:** The modified KS.test is done correctly, but it still does not reject incorrect normal assumption. The MD plot is needed.

---

```
x = model_gamma$residuals
s = sd(x)
t = ks.test(x, "pgamma", 0, s)$p
n = length(Xi)
set.seed(42)
u = rep(0,1000)
for (i in 1:1000) {
  z = rnorm(n, 0, s)
  x = lm(z ~ 1)$resid
  v = sd(x)
  u[i] = ks.test(z, "pgamma", 0, v)$p.value
}
proportion = mean(u < t)
proportion
```

```
## [1] 0
```

---

**Teacher:** The modified KS.test against Gamma is not correctly implemented.

Thus it rejects correct Gamma distribution.

One should use x=model_normal$residuals+ model_normal_coef[1] i.e. use W=Y-X$\beta$

instead of x=model_gamma$residuals

Also, the codes in the loop need to be revised correspondingly.

Finally, the MD plot has not been implemented.

---

2. Use data B in Example 1 (around pages 25-28) to mimic the analysis in Example 1 (testing Ho: $\mu = 10$)

```
B = c(14.0, 8.8, 11.2, 14.2, 11.8, 6.4, 9.8, 11.3, 9.3, 13.6)
```

```
qqPlot(B, dist= "norm")
```

```
## [1] 6 4
```

```
x = B[2*(1:5)]
y = B[-2*(1:5)]
```

```
library(coin)
```

```
## Warning: package 'coin' was built under R version 4.3.2
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'coin'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     alpha
```

```
independence_test(x~y)
```

```
##
##  Asymptotic General Independence Test
##
## data:  x by y
## Z = -1.1727, p-value = 0.2409
## alternative hypothesis: two.sided
```

```
t.test(B, mu = 10)
```

```
##
##  One Sample t-test
##
## data:  B
## t = 1.3059, df = 9, p-value = 0.224
## alternative hypothesis: true mean is not equal to 10
## 95 percent confidence interval:
##   9.238399 12.841601
## sample estimates:
## mean of x
##     11.04
```

```
stem(B)
```

```
##
##   The decimal point is at the |
##
##    6 | 4
##    8 | 838
##   10 | 238
##   12 | 6
##   14 | 02
```

```
wilcox.test(B, mu = 10)
```

```
## Warning in wilcox.test.default(B, mu = 10): cannot compute exact p-value with
## ties
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  B
## V = 41, p-value = 0.1846
## alternative hypothesis: true location is not equal to 10
```

Report

Question 1

In question 1 we create data via the addition of a sample of binomial data and a sample of gamma distributed data, from here we create two models to try and explain the relationship between Y and X, one is a normal model and one is a gamma model. Fitting confidence bands on the two QQ-plots, we can see that only in the Normal Plot does the confidence bands encapsulate all the data, indicating this likely where the distribution is coming from, which is interesting since we didn't use normal data. From here we create MD plots to view

the ecdf of the data vs the ecf of the fitted model. For some reason, in both models, the fitted values look close to the actual data, which is interesting, though in the normal model it does look closer as the fitted data does not ever fall outside the dotted lines. Finally, using the modified ks.test to see if the residuals of the model are normally distributed, we can see in the normal case they are, with a p-value of .268 whereas in the gamma model they do not seem to be normally distributed, with a p-value of 0.

Question 2

In question 2 we are mirroring the analysis from shoe data A but on shoe data 2. Based on the independence test we can say that the data does seem independent. From here, all of the tests, t.test and the non-parametric Wilcox test, suggest that the true mean is not different than 10. We trust the t.test more here as we can see from the QQplot that the data is approximately normally distributed.

---

**Teacher:** Recall the remark around page 23.

**Remark.** The idea of the MD plot will be discussed later on.

**The previous MD plot codes is to implement following steps:**

b1. Suppose that the data fit the model $Y_i = \beta X_i + W_i$, where $W_i$'s are i.i.d. $\sim N(\mu, \sigma^2)$. Moreover, $X_i$'s are dense around $x_o$. Let $m = ||\{X_i : \ |X_i - x_o| < \delta_n\}||$ for some $\delta_n > 0$. By making a transformation $X_i^* = X_i - x_o$ and $W_i^* = W_i + \beta x_o$, WLOG, we can assume $x_o = 0$.

b2. Obtain $\hat{\beta}$, the LSE of $\beta$ based on $(X_i, Y_i)$'s.

b3. Take a random sample of size $m$ from the $X_i$'s in a neighborhood of $x_o$ (before $X_i - x_o$), *i.e.*, $|X_i - x_o| \leq \delta_n$, or a neighborhood of $\mathbf{0}$ (after $X_i - x_o$), where $m$ and $\delta_n$ are as in (b1), take another random sample of size $n - m$ from the $X_i$'s satisfying $|X_i - x_o| > \delta_n$, and take a random sample of size $n$ from $Y_i$'s satisfying $|X_i - x_o| \leq \delta_n$.

**Additional comments regarding $X \sim$ binomial(10,0.1) in this homework:**

> dbinom(0:10,10,0.1)

[1] 0.3486784401 0.3874204890 0.1937102445 0.0573956280 0.0111602610

[6] 0.0014880348 0.0001377810 0.0000087480 0.0000003645 0.0000000090

[11] 0.0000000001

$x_o = ?$

$\delta_n = ?$

yields a sample of $(X, W)$'s, say $(X_1^*, W_i^*)$, $i = 1, ..., n$.

(Note that $Y = \beta X + W = W$ if $X = 0$.)

b4. Let $Y_i^* = \hat{\beta} X_i^* + W_i^*$.

---

**Homework 5.1.2.** Derive the most powerful test $\psi_2$ of size 0.05 for Case 2 if $X_1, ..., X_{10}$ are i.i.d. from bin(1,p) **(instead of bin(n,p))**. What is the probability of type I or II error for $p \in \{0.3, 0.9\}$, and the size of the test $\psi_2$ ?

**Hint:** Check 448 [21].

5.1.2 In this test we reject if the number of successes is less than or equal to 2 and greater than or equal to 8. The test has size .05, the type 2 error probabilities for .3 and .9 are .38 and .93

---

**Teacher:** The answer is incorrect. Your test is not MP test and your answers to the other questions in this problem are wrong too. Your test is level 0.05 test but not size 0.05.

From 448 and 502:

$\alpha$ – size of the test defined by $\alpha = \sup_{\theta \in \Theta_o} E_\theta(\phi)$.

448 [21] The MP test for $H_o$: $\theta = \theta_o$ v.s. $H_a$: $\theta = \theta_a$, the MP test has the RR satisfying: $\frac{L(\theta_o)}{L(\theta_a)}$____$k$ and $P_\theta(RR) = \alpha$ if $\theta =$ ____. **key:** $\leq$, $\underline{\theta_o}$,

pbinom(0:10,10,0.5)

[1] 0.0009765625 0.0107421875 0.0546875000 0.1718750000 0.3769531250

[6] 0.6230468750 0.8281250000 0.9453125000 0.9892578125 0.9990234375

[11] 1.0000000000

$\frac{L(\theta_o)}{L(\theta_a)} < k$

The test is $\phi = I(\frac{L(\theta_o)}{L(\theta_a)} < k) + cI(\frac{L(\theta_o)}{L(\theta_a)} = k)$ such that $E(\phi) = 0.05$.

You need to find k and c !!!

---