Statistics 3080
Homework 07
Rahul Zalkikar

Problem 01

```
>   set.seed(5041998)
>
>   K <- 10000
>   alpha = 0.05
>   samp.8 <- replicate(K, rnorm(8, mean = 121.8, sd = 34.7))
>
>   mean.8 <- apply(samp.8,2,mean)
>   test_stat.8 <- (mean.8 - 121.8)/(34.7/sqrt(8))
>   p_value.8 <- pnorm(-abs(test_stat.8))
>   Reject_Null.8 <- as.logical(2*p_value.8<=alpha)
>   samp.8_vect <- as.integer(Reject_Null.8)
>
>   samp.8_sum <- sum(samp.8_vect)
>   Prop_Reject_Null.8.1 <- samp.8_sum / 10000
>
>   samp.24 <- replicate(K, rnorm(24, mean = 121.8, sd = 34.7))
>
>   mean.24 <- apply(samp.24,2,mean)
>   test_stat.24 <- (mean.24 - 121.8)/(34.7/sqrt(24))
>   p_value.24 <- pnorm(-abs(test_stat.24))
>   Reject_Null.24 <- as.logical(2*p_value.24<=alpha)
>   samp.24_vect <- as.integer(Reject_Null.24)
>
>   samp.24_sum <- sum(samp.24_vect)
>   Prop_Reject_Null.24.1 <- samp.24_sum / 10000
>
>   samp.48 <- replicate(K, rnorm(48, mean = 121.8, sd = 34.7))
>
>   mean.48 <- apply(samp.48,2,mean)
>   test_stat.48 <- (mean.48 - 121.8)/(34.7/sqrt(48))
>   p_value.48 <- pnorm(-abs(test_stat.48))
>   Reject_Null.48 <- as.logical(2*p_value.48<=alpha)
>   samp.48_vect <- as.integer(Reject_Null.48)
>
>   samp.48_sum <- sum(samp.48_vect)
>   Prop_Reject_Null.48.1 <- samp.48_sum / 10000
>
>   Prop_Reject_Null.8.1
```

```
[1] 0.0489
>   Prop_Reject_Null.24.1
[1] 0.0454
>   Prop_Reject_Null.48.1
[1] 0.05
```

Problem 02

```
>   K <- 10000
>   alpha = 0.05
>   samp.8 <- replicate(K, rnorm(8, mean = 121.8, sd = 34.7))
>
>   mean.8 <- apply(samp.8,2,mean)
>   sd.8 <- apply(samp.8,2,sd)
>   test_stat.8 <- (mean.8 - 121.8)/(sd.8/sqrt(8))
>   p_value.8 <- pt(-abs(test_stat.8), 7)
>   Reject_Null.8 <- as.logical(2*p_value.8<=alpha)
>   samp.8_vect <- as.integer(Reject_Null.8)
>
>   samp.8_sum <- sum(samp.8_vect)
>   Prop_Reject_Null.8.2 <- samp.8_sum / 10000
>
>   samp.24 <- replicate(K, rnorm(24, mean = 121.8, sd = 34.7))
>
>   mean.24 <- apply(samp.24,2,mean)
>   sd.24 <- apply(samp.24,2,sd)
>   test_stat.24 <- (mean.24 - 121.8)/(sd.24/sqrt(24))
>   p_value.24 <- pt(-abs(test_stat.24), 23)
>   Reject_Null.24 <- as.logical(2*p_value.24<=alpha)
>   samp.24_vect <- as.integer(Reject_Null.24)
>
>   samp.24_sum <- sum(samp.24_vect)
>   Prop_Reject_Null.24.2 <- samp.24_sum / 10000
>
>   samp.48 <- replicate(K, rnorm(48, mean = 121.8, sd = 34.7))
>
>   mean.48 <- apply(samp.48,2,mean)
>   sd.48 <- apply(samp.48,2,sd)
>   test_stat.48 <- (mean.48 - 121.8)/(sd.48/sqrt(48))
>   p_value.48 <- pt(-abs(test_stat.48), 47)
>   Reject_Null.48 <- as.logical(2*p_value.48<=alpha)
>   samp.48_vect <- as.integer(Reject_Null.48)
>
>   samp.48_sum <- sum(samp.48_vect)
>   Prop_Reject_Null.48.2 <- samp.48_sum / 10000
```

```
>
>   Prop_Reject_Null.8.2
[1] 0.0546
>   Prop_Reject_Null.24.2
[1] 0.0475
>   Prop_Reject_Null.48.2
[1] 0.0526
```

Problem 03

```
>   K <- 10000
>   alpha = 0.05
>   samp.8 <- replicate(K, rnorm(8, mean = 121.8, sd = 34.7))
>
>   mean.8 <- apply(samp.8,2,mean)
>   sd.8 <- apply(samp.8,2,sd)
>   test_stat.8 <- (mean.8 - 121.8)/(sd.8/sqrt(8))
>   p_value.8 <- pnorm(-abs(test_stat.8))
>   Reject_Null.8 <- as.logical(2*p_value.8<=alpha)
>   samp.8_vect <- as.integer(Reject_Null.8)
>
>   samp.8_sum <- sum(samp.8_vect)
>   Prop_Reject_Null.8.3 <- samp.8_sum / 10000
>
>   samp.24 <- replicate(K, rnorm(24, mean = 121.8, sd = 34.7))
>
>   mean.24 <- apply(samp.24,2,mean)
>   sd.24 <- apply(samp.24,2,sd)
>   test_stat.24 <- (mean.24 - 121.8)/(sd.24/sqrt(24))
>   p_value.24 <- pnorm(-abs(test_stat.24))
>   Reject_Null.24 <- as.logical(2*p_value.24<=alpha)
>   samp.24_vect <- as.integer(Reject_Null.24)
>
>   samp.24_sum <- sum(samp.24_vect)
>   Prop_Reject_Null.24.3 <- samp.24_sum / 10000
>
>   samp.48 <- replicate(K, rnorm(48, mean = 121.8, sd = 34.7))
>
>   mean.48 <- apply(samp.48,2,mean)
>   sd.48 <- apply(samp.48,2,sd)
>   test_stat.48 <- (mean.48 - 121.8)/(sd.48/sqrt(48))
>   p_value.48 <- pnorm(-abs(test_stat.48))
>   Reject_Null.48 <- as.logical(2*p_value.48<=alpha)
>   samp.48_vect <- as.integer(Reject_Null.48)
>
```

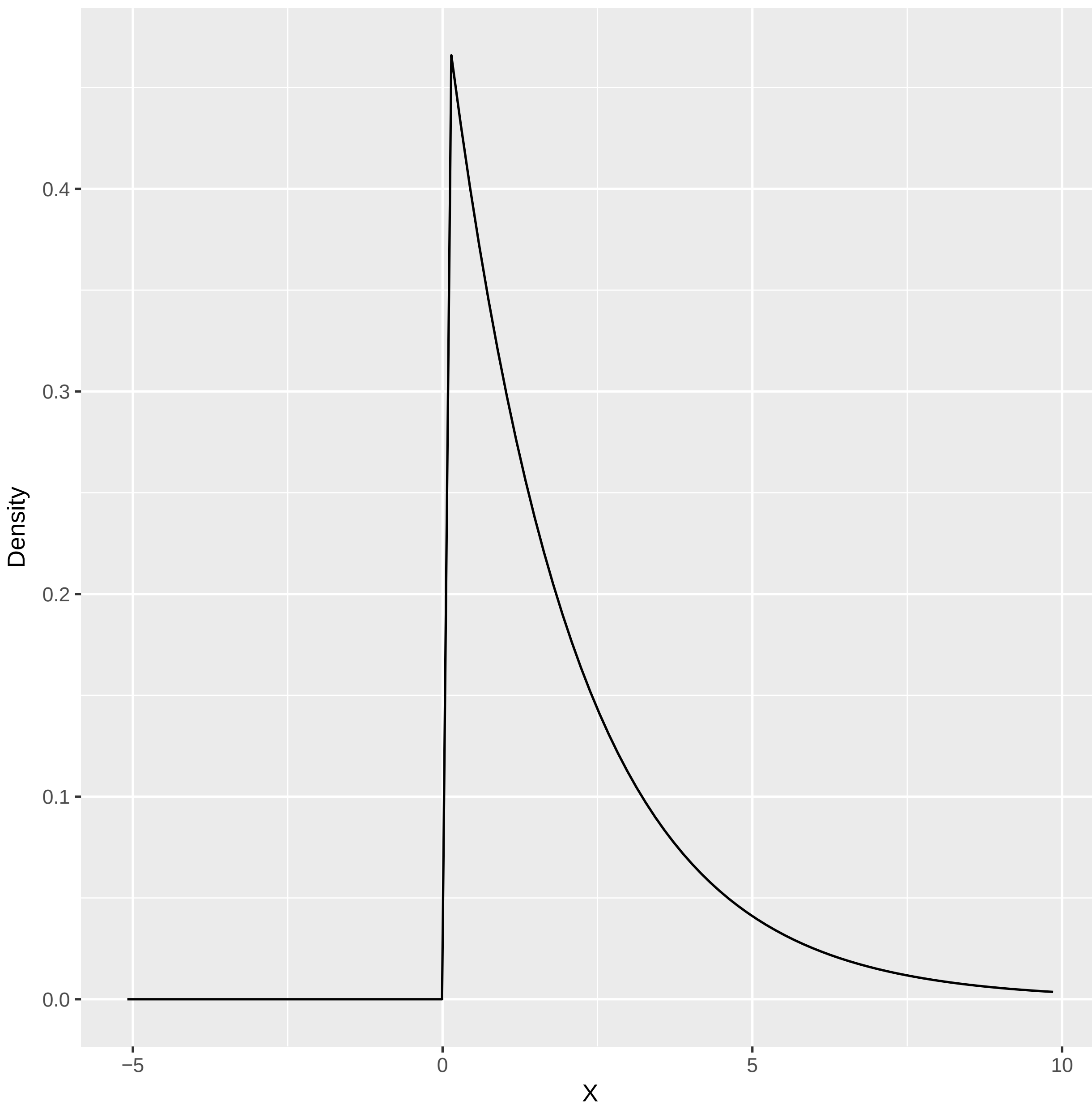
```
>   samp.48_sum <- sum(samp.48_vect)
>   Prop_Reject_Null.48.3 <- samp.48_sum / 10000
>
>   Prop_Reject_Null.8.3
[1] 0.0947
>   Prop_Reject_Null.24.3
[1] 0.0608
>   Prop_Reject_Null.48.3
[1] 0.0597
```

Problem 04

```
>   results <- matrix(c(Prop_Reject_Null.8.1, Prop_Reject_Null.8.2,
+                       Prop_Reject_Null.8.3, Prop_Reject_Null.24.1,
+                       Prop_Reject_Null.24.2, Prop_Reject_Null.24.3,
+                       Prop_Reject_Null.48.1, Prop_Reject_Null.48.2,
+                       Prop_Reject_Null.48.3),
+                     ncol=3,byrow=TRUE)
>   colnames(results) <- c("Prob1: Type1Error","Prob2: Type1Error","Prob3: Type1Error")
>   rownames(results) <- c("Sample Size 8", "Sample Size 24", "Sample Size 48")
>   results <- as.table(results)
>   results
               Prob1: Type1Error Prob2: Type1Error Prob3: Type1Error
Sample Size 8             0.0489            0.0546            0.0947
Sample Size 24            0.0454            0.0475            0.0608
Sample Size 48            0.0500            0.0526            0.0597
>   # Conclusions: The Type 1 Error is closest to the desired 0.05 when the population
>   # standard deviation is known and a one-sample z-test is conducted for a normal
>   # distribution. When the population standard deviation is unkown for a normal
>   # distribution and a one-sample t-test is used, the proportion of incorrectly
>   # rejected null hypothesis increases substantially. When the population standard
>   # deviation is unkown for a normal distribution and a one-sample z test is used,
>   # the proportion of incorrectly rejected null hypothesis also increases
>   # substantially. This leads to the conclusion that knowing the population standard
>   # deviation (while using a one-sample z-test) is most effective in limiting Type 1
>   # Errors to an acceptable proportion.
```

Problem 05

```
>   #Part A
>   library(ggplot2)
>   r <- rnorm(10000,mean=2,sd=2)
>   Xdata2 <- data.frame(X=r)
>   dist2 <- ggplot(Xdata2, aes(x=X))
>   dist2 + stat_function(fun=dchisq, args=list(df=2)) +ylab("Density")
```




```
>   # The Density curve is similar in shape to the first quadrant shape of a
>   # 1/(x^2) graph for x>0. y=0 for all x<0.
>
>   #Part B
>   K <- 10000
>   alpha = 0.05
>   samp.8 <- replicate(K, rchisq(8,df=2))
>
>   mean.8 <- apply(samp.8,2,mean)
>   test_stat.8 <- (mean.8 - 2)/(2/sqrt(8))
```

```
>   p_value.8 <- pnorm(-abs(test_stat.8))
>   Reject_Null.8 <- as.logical(2*p_value.8<=alpha)
>   samp.8_vect <- as.integer(Reject_Null.8)
>
>   samp.8_sum <- sum(samp.8_vect)
>   Prop_Reject_Null.8.1 <- samp.8_sum / 10000
>
>   samp.24 <- replicate(K, rchisq(24,df=2))
>
>   mean.24 <- apply(samp.24,2,mean)
>   test_stat.24 <- (mean.24 - 2)/(2/sqrt(24))
>   p_value.24 <- pnorm(-abs(test_stat.24))
>   Reject_Null.24 <- as.logical(2*p_value.24<=alpha)
>   samp.24_vect <- as.integer(Reject_Null.24)
>
>   samp.24_sum <- sum(samp.24_vect)
>   Prop_Reject_Null.24.1 <- samp.24_sum / 10000
>
>   samp.48 <- replicate(K, rchisq(48,df=2))
>
>   mean.48 <- apply(samp.48,2,mean)
>   test_stat.48 <- (mean.48 - 2)/(2/sqrt(48))
>   p_value.48 <- pnorm(-abs(test_stat.48))
>   Reject_Null.48 <- as.logical(2*p_value.48<=alpha)
>   samp.48_vect <- as.integer(Reject_Null.48)
>
>   samp.48_sum <- sum(samp.48_vect)
>   Prop_Reject_Null.48.1 <- samp.48_sum / 10000
>
>   Prop_Reject_Null.8.1
[1] 0.0451
>   Prop_Reject_Null.24.1
[1] 0.0466
>   Prop_Reject_Null.48.1
[1] 0.0494
>   #Part C
>   K <- 10000
>   alpha = 0.05
>   samp.8 <- replicate(K, rchisq(8,df=2))
>
>   mean.8 <- apply(samp.8,2,mean)
>   sd.8 <- apply(samp.8,2,sd)
>   test_stat.8 <- (mean.8 - 2)/(sd.8/sqrt(8))
>   p_value.8 <- pt(-abs(test_stat.8), 7)
```

```
>   Reject_Null.8 <- as.logical(2*p_value.8<=alpha)
>   samp.8_vect <- as.integer(Reject_Null.8)
>
>   samp.8_sum <- sum(samp.8_vect)
>   Prop_Reject_Null.8.2 <- samp.8_sum / 10000
>
>   samp.24 <- replicate(K, rchisq(24,df=2))
>
>   mean.24 <- apply(samp.24,2,mean)
>   sd.24 <- apply(samp.24,2,sd)
>   test_stat.24 <- (mean.24 - 2)/(sd.24/sqrt(24))
>   p_value.24 <- pt(-abs(test_stat.24), 23)
>   Reject_Null.24 <- as.logical(2*p_value.24<=alpha)
>   samp.24_vect <- as.integer(Reject_Null.24)
>
>   samp.24_sum <- sum(samp.24_vect)
>   Prop_Reject_Null.24.2 <- samp.24_sum / 10000
>
>   samp.48 <- replicate(K, rchisq(48,df=2))
>
>   mean.48 <- apply(samp.48,2,mean)
>   sd.48 <- apply(samp.48,2,sd)
>   test_stat.48 <- (mean.48 - 2)/(sd.48/sqrt(48))
>   p_value.48 <- pt(-abs(test_stat.48), 47)
>   Reject_Null.48 <- as.logical(2*p_value.48<=alpha)
>   samp.48_vect <- as.integer(Reject_Null.48)
>
>   samp.48_sum <- sum(samp.48_vect)
>   Prop_Reject_Null.48.2 <- samp.48_sum / 10000
>
>   Prop_Reject_Null.8.2
[1] 0.108
>   Prop_Reject_Null.24.2
[1] 0.0757
>   Prop_Reject_Null.48.2
[1] 0.0651
>   #Part D
>   K <- 10000
>   alpha = 0.05
>   samp.8 <- replicate(K, rchisq(8,df=2))
>
>   mean.8 <- apply(samp.8,2,mean)
>   sd.8 <- apply(samp.8,2,sd)
>   test_stat.8 <- (mean.8 - 2)/(sd.8/sqrt(8))
```

```
>   p_value.8 <- pnorm(-abs(test_stat.8))
>   Reject_Null.8 <- as.logical(2*p_value.8<=alpha)
>   samp.8_vect <- as.integer(Reject_Null.8)
>
>   samp.8_sum <- sum(samp.8_vect)
>   Prop_Reject_Null.8.3 <- samp.8_sum / 10000
>
>   samp.24 <- replicate(K, rchisq(24,df=2))
>
>   mean.24 <- apply(samp.24,2,mean)
>   sd.24 <- apply(samp.24,2,sd)
>   test_stat.24 <- (mean.24 - 2)/(sd.24/sqrt(24))
>   p_value.24 <- pnorm(-abs(test_stat.24))
>   Reject_Null.24 <- as.logical(2*p_value.24<=alpha)
>   samp.24_vect <- as.integer(Reject_Null.24)
>
>   samp.24_sum <- sum(samp.24_vect)
>   Prop_Reject_Null.24.3 <- samp.24_sum / 10000
>
>   samp.48 <- replicate(K, rchisq(48,df=2))
>
>   mean.48 <- apply(samp.48,2,mean)
>   sd.48 <- apply(samp.48,2,sd)
>   test_stat.48 <- (mean.48 - 2)/(sd.48/sqrt(48))
>   p_value.48 <- pnorm(-abs(test_stat.48))
>   Reject_Null.48 <- as.logical(2*p_value.48<=alpha)
>   samp.48_vect <- as.integer(Reject_Null.48)
>
>   samp.48_sum <- sum(samp.48_vect)
>   Prop_Reject_Null.48.3 <- samp.48_sum / 10000
>
>   Prop_Reject_Null.8.3
[1] 0.1402
>   Prop_Reject_Null.24.3
[1] 0.0898
>   Prop_Reject_Null.48.3
[1] 0.0748
```

Problem 06

```
>   results <- matrix(c(Prop_Reject_Null.8.1, Prop_Reject_Null.8.2,
+                     Prop_Reject_Null.8.3, Prop_Reject_Null.24.1,
+                     Prop_Reject_Null.24.2, Prop_Reject_Null.24.3,
+                     Prop_Reject_Null.48.1, Prop_Reject_Null.48.2,
```

```
+                         Prop_Reject_Null.48.3),
+                       ncol=3,byrow=TRUE)
>   colnames(results) <- c("Prob1: Type1Error","Prob2: Type1Error","Prob3: Type1Error")
>   rownames(results) <- c("Sample Size 8", "Sample Size 24", "Sample Size 48")
>   results <- as.table(results)
>   results
               Prob1: Type1Error Prob2: Type1Error Prob3: Type1Error
Sample Size 8             0.0451            0.1080            0.1402
Sample Size 24            0.0466            0.0757            0.0898
Sample Size 48            0.0494            0.0651            0.0748
>   # Conclusions: The Type 1 Error is closest to the desired 0.05 when the population
>   # standard deviation is known and a one-sample z-test is conducted for a chi
>   # square distribution. When the population standard deviation is unkown for a
>   # chi square distribution and a one-sample t-test is used, the proportion of
>   # incorrectly rejected null hypothesis increases substantially. When the
>   # population standard deviation is unkown for a chi square distribution and a
>   # one-sample z test is used, the proportion of incorrectly rejected null
>   # hypothesis also increases substantially. This leads to the conclusion that
>   # knowing the population standard deviation (while using a one-sample z-test) is
>   # most effective in limiting Type 1 Errors to an acceptable proportion.
```

Problem 07

```
>   # Overall Conclusions: Since the significance level is set to 0.05 (5%), it is
>   # acceptable to have a 5% probability of incorrectly rejecting the null
>   # hypothesis, or a Type 1 Error. The Type 1 Error is closest to the desired 0.05
>   # when the population standard deviation is known and a one-sample z-test is
>   # conducted for a chi square distribution. The Type 1 Error is closest to the
>   # desired 0.05 when the population standard deviation is known and a one-sample
>   # z-test is conducted for a normal distribution. When the population standard
>   # deviation is unkown for a chi square distribution and a one-sample t-test is
>   # used, the proportion of incorrectly rejected null hypothesis increases
>   # substantially. When the population standard deviation is unkown for a chi
>   # square distribution and a one-sample z test is used, the proportion of
>   # incorrectly rejected null hypothesis also increases substantially. The same
>   # trend applies to a normal distribution, although differences between the
>   # proportions of incorrectly rejected null hypothesis across test-scenarios is
>   # less when using a normal distribution. This affirms the notion that knowing
>   # the population standard deviation is most effective in limiting Type 1 Errors
>   # to an acceptable proportion.
```

References:

- Simulations in R.R

- https://www.rdocumentation.org/packages/base/versions/3.4.3/topics/integer