[1] Review of HW3 (Periodic Review Policy)

- Check the data!
- Use function and do not use the same syntax repeatedly.
- Again, review your answer throughly and compare it with the solution code.

```
df_product1 <- readxl::read_excel(here::here("data/Homework3_data.xlsx"), sheet = "Product1")
df_product2 <- readxl::read_excel(here::here("data/Homework3_data.xlsx"), sheet = "Product2")
df_product3 <- readxl::read_excel(here::here("data/Homework3_data.xlsx"), sheet = "Product3")
```

Check the Data First!

```
summary(df_product1)
```

```
##     Region1         Region2         Region3         Region4
##  Min.   :17.19   Min.   :15.61   Min.   :15.25   Min.   :16.92
##  1st Qu.:26.63   1st Qu.:26.49   1st Qu.:26.91   1st Qu.:26.28
##  Median :29.90   Median :29.80   Median :30.04   Median :29.78
##  Mean   :29.88   Mean   :29.83   Mean   :30.16   Mean   :29.67
##  3rd Qu.:32.73   3rd Qu.:33.29   3rd Qu.:33.47   3rd Qu.:33.04
##  Max.   :46.19   Max.   :46.13   Max.   :45.29   Max.   :47.50
```

```
summary(df_product2)
```

```
##     Region1          Region2          Region3          Region4
##  Min.   : 1.922   Min.   : 2.438   Min.   : 2.142   Min.   : 1.976
##  1st Qu.: 8.151   1st Qu.: 8.028   1st Qu.: 8.044   1st Qu.: 8.338
##  Median :10.318   Median :10.314   Median : 9.900   Median :10.470
##  Mean   :10.289   Mean   :10.168   Mean   : 9.891   Mean   :10.335
##  3rd Qu.:12.553   3rd Qu.:12.291   3rd Qu.:11.935   3rd Qu.:12.199
##  Max.   :20.648   Max.   :18.508   Max.   :18.052   Max.   :18.802
```

```
summary(df_product3)
```

```
##     Region1          Region2          Region3          Region4
##  Min.   : 1.370   Min.   : 1.348   Min.   :-1.009   Min.   : 1.065
##  1st Qu.: 7.602   1st Qu.: 7.560   1st Qu.: 7.766   1st Qu.: 7.572
##  Median : 9.765   Median : 9.592   Median : 9.683   Median : 9.684
##  Mean   : 9.781   Mean   : 9.675   Mean   : 9.812   Mean   : 9.889
##  3rd Qu.:12.039   3rd Qu.:11.788   3rd Qu.:12.134   3rd Qu.:11.851
##  Max.   :19.203   Max.   :18.210   Max.   :18.559   Max.   :18.576
```

Setting

```
LT <- 11
T_SL <- 0.95
```

Analysis

- You do the same calculations multiple times.
- Highly recommend to use `function` to avoid bugs and having longer code.

Define Function

```
calculate_inventory_cost <- function(T_SL, ShipCost, df_site) {
  LT_demand <- RcppRoll::roll_sum(df_site, LT)
  OUL <- as.numeric(quantile(LT_demand, T_SL))
  SS <- OUL - LT * mean(df_site)
  OrderQ <- 6 * mean(df_site)
  CycleStock <- OrderQ / 2
  Inventory <- SS + CycleStock
  HoldingCost <- Inventory * 0.15
  ShippingCost <- mean(df_site) * ShipCost

  output <- list(LT_demand = LT_demand,
                 OUL = OUL,
                 SS = SS,
                 OrderQ = OrderQ,
                 CycleStock = CycleStock,
                 Inventory = Inventory,
                 HoldingCost = HoldingCost,
                 ShippingCost = ShippingCost)
  return(output)
}
```

Run for each product

```
# product 1
Region1 <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.19, df_site = df_product1$Region1)
Region2 <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.19, df_site = df_product1$Region2)
Region3 <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.19, df_site = df_product1$Region3)
Region4 <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.19, df_site = df_product1$Region4)
Central <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.29, df_site = rowSums(df_product1))

sep_TC <- Region1$HoldingCost + Region2$HoldingCost + Region3$HoldingCost + Region4$HoldingCost +
  Region1$ShippingCost + Region2$ShippingCost + Region3$ShippingCost + Region4$ShippingCost
sep_TC
```

```
## [1] 93.34966
```

```
cent_TC <- Central$HoldingCost + Central$ShippingCost
cent_TC
```

```
## [1] 98.6451
```

```
prod1 <- c(sep_TC, cent_TC)
```

```r
# product 2
Region1 <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.19, df_site = df_product2$Region1)
Region2 <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.19, df_site = df_product2$Region2)
Region3 <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.19, df_site = df_product2$Region3)
Region4 <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.19, df_site = df_product2$Region4)
Central <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.29, df_site = rowSums(df_product2))

sep_TC <- Region1$HoldingCost + Region2$HoldingCost + Region3$HoldingCost + Region4$HoldingCost +
  Region1$ShippingCost + Region2$ShippingCost + Region3$ShippingCost + Region4$ShippingCost
sep_TC
```

```
## [1] 36.66653
```

```r
cent_TC <- Central$HoldingCost + Central$ShippingCost
cent_TC
```

```
## [1] 35.21223
```

```r
prod2 <- c(sep_TC, cent_TC)
```

```r
# product 3
Region1 <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.19, df_site = df_product3$Region1)
Region2 <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.19, df_site = df_product3$Region2)
Region3 <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.19, df_site = df_product3$Region3)
Region4 <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.19, df_site = df_product3$Region4)
Central <- calculate_inventory_cost(T_SL = T_SL, ShipCost = 0.29, df_site = rowSums(df_product3))

sep_TC <- Region1$HoldingCost + Region2$HoldingCost + Region3$HoldingCost + Region4$HoldingCost +
  Region1$ShippingCost + Region2$ShippingCost + Region3$ShippingCost + Region4$ShippingCost
sep_TC
```

```
## [1] 34.99976
```

```r
cent_TC <- Central$HoldingCost + Central$ShippingCost
cent_TC
```

```
## [1] 36.59612
```

```r
prod3 <- c(sep_TC, cent_TC)
```

Summary Table

```
tb <- rbind(t(prod1), t(prod2), t(prod3))
rownames(tb) <- c("product1", "product2", "product3")
colnames(tb) <- c("Separate", "Centralized")
kableExtra::kable(tb)
```

	Separate	Centralized
product1	93.34966	98.64510
product2	36.66653	35.21223
product3	34.99976	36.59612

When to centralize?

- Think about what kills the benefit of centralization.
  - Lower demand uncertainty
  - Demand in different areas are interdependent.
- Look at the data and compare the mean and SD.
  - If the SD is high, you may tempt to centralize.
  - But, you also need to take care of the mean.
- Look at the correlation of demand across regions.

Solution Key:

- If CV is high, it is a slow-selling item, so better to put it far from customers.
- If cor between products is high, the benefit of centralization cancels out.

Product 1

```
apply(df_product1, 2, mean)
```

```
##  Region1  Region2  Region3  Region4
## 29.87545 29.83370 30.15583 29.66605
```

```
apply(df_product1, 2, sd)
```

```
##  Region1  Region2  Region3  Region4
## 4.896402 5.254902 4.936193 4.949797
```

```
apply(df_product1, 2, EnvStats::cv) # coefficient of variation
```

```
##   Region1   Region2   Region3   Region4
## 0.1638938 0.1761398 0.1636895 0.1668506
```

```
cor(df_product1) # correlation matrix
```

```
##             Region1    Region2      Region3      Region4
## Region1  1.00000000 0.01880890  0.061522460 -0.039194126
## Region2  0.01880890 1.00000000  0.039607139  0.069235453
## Region3  0.06152246 0.03960714  1.000000000 -0.008223714
## Region4 -0.03919413 0.06923545 -0.008223714  1.000000000
```

Product 2

```
apply(df_product2, 2, mean)
```

```
##   Region1   Region2   Region3   Region4
## 10.289044 10.167527  9.891387 10.335439
```

```
apply(df_product2, 2, sd)
```

```
##  Region1  Region2  Region3  Region4
## 2.945669 2.994597 2.922232 2.862480
```

```
apply(df_product2, 2, EnvStats::cv) # coefficient of variation
```

```
##   Region1   Region2   Region3   Region4
## 0.2862918 0.2945256 0.2954319 0.2769578
```

```
cor(df_product2) # correlation matrix
```

```
##             Region1    Region2     Region3     Region4
## Region1  1.00000000 0.01863696 -0.03411111 -0.03482577
## Region2  0.01863696 1.00000000  0.01007484  0.02120485
## Region3 -0.03411111 0.01007484  1.00000000  0.03498528
## Region4 -0.03482577 0.02120485  0.03498528  1.00000000
```

Product 3

```
apply(df_product3, 2, mean)
```

```
##  Region1  Region2  Region3  Region4
## 9.781189 9.675070 9.812385 9.889413
```

```
apply(df_product3, 2, sd)
```

```
##  Region1  Region2  Region3  Region4
## 3.109725 3.065133 3.400357 3.193302
```

```
apply(df_product3, 2, EnvStats::cv) # coefficient of variation
```

```
##   Region1   Region2   Region3   Region4
## 0.3179291 0.3168073 0.3465373 0.3229011
```

```
cor(df_product3) # correlation matrix
```

```
##           Region1   Region2   Region3   Region4
## Region1 1.0000000 0.5655619 0.5403112 0.5238858
## Region2 0.5655619 1.0000000 0.5243354 0.4777590
## Region3 0.5403112 0.5243354 1.0000000 0.5550859
## Region4 0.5238858 0.4777590 0.5550859 1.0000000
```