# Programming Of Data Science

## DECLARATION

we the authors of this work, verify that:

- We hold a copy of this assignment that we can produce if the original is lost or damaged.

  Weherebycertifythatnopartofthisassignment/producthasbeencopiedfromanyotherstudent's work or from any other source except where due acknowledgement is made in the assignment.

- No part of this assignment/product has been written/produced for us by another person except where such collaboration has been authorised by the subject lecturer/tutor concerned.

- Weareawarethatthisworkmaybereproducedandsubmittedtoplagiarismdetectionsoftwarepro- grams for the purpose of detecting possible plagiarism (which may retain a copy on its database for future plagiarism checking).

- We hereby certify that we have read and understand what the School of Computing, Engineering and Mathematics defines as minor and substantial breaches of misconduct as outlined in the learning guide for this unit.

**Installing all required libraries**

```
library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ lubridate 1.9.2     ✓ tibble    3.2.1
```

```
## ✓ purrr     1.0.1     ✓ tidyr     1.3.0
## ✓ readr     2.1.4

## ── Conflicts ─────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force
all conflicts to become errors
```

## Question 1

### For one of the data sets, write the code to compute the total revenue of each store at the end of each day.

### Is there a noted difference between the days?

### Write also the code to calculate the total revenue over the seven day period. Plot the latter on a graph.

### Compare the revenue between the two data sets, is there a difference?

To do this question, let's create data frames for both the sales data sets.

```
# using read.csv() function, we are locating the target file and saving it
into variables 'sales1' and 'sales2' as Data Frames.

sales1 <- read.csv("Datasets/sales_pg_1.csv")
sales2 <- read.csv("Datasets/sales_pg_2.csv")

# Print out first 6 rows of both data frames.

head(sales1)
```

```
##   product_id store_id       date sales revenue stock price promo_type_1
## 1      P0001    S0001 2018-01-28     0       0    10  6.75         PR14
## 2      P0001    S0002 2018-01-28     0       0     8  6.75         PR14
## 3      P0001    S0004 2018-01-28     0       0     7  6.75         PR14
## 4      P0001    S0008 2018-01-28     0       0     6  6.75         PR14
## 5      P0001    S0012 2018-01-28     0       0     7  6.75         PR14
## 6      P0001    S0013 2018-01-28     0       0    10  6.75         PR14
##   promo_bin_1 promo_discount_2 promo_discount_type_2
## 1                          NA                    NA
## 2                          NA                    NA
## 3                          NA                    NA
## 4                          NA                    NA
```

```
## 5                           NA                    NA
## 6                           NA                    NA

head(sales2)

##   product_id store_id       date sales revenue stock price promo_type_1
## 1      P0001    S0001 2018-04-28     0       0     5   7.9         PR14
## 2      P0001    S0002 2018-04-28     0       0     4   7.9         PR14
## 3      P0001    S0004 2018-04-28     0       0     4   7.9         PR14
## 4      P0001    S0008 2018-04-28     0       0    10   7.9         PR14
## 5      P0001    S0012 2018-04-28     0       0     5   7.9         PR14
## 6      P0001    S0013 2018-04-28     0       0     1   7.9         PR14
##   promo_bin_1 promo_discount_2 promo_discount_type_2
## 1                           NA                    NA
## 2                           NA                    NA
## 3                           NA                    NA
## 4                           NA                    NA
## 5                           NA                    NA
## 6                           NA                    NA
```

### A) write the code to compute the total revenue of each store at the end of each day.

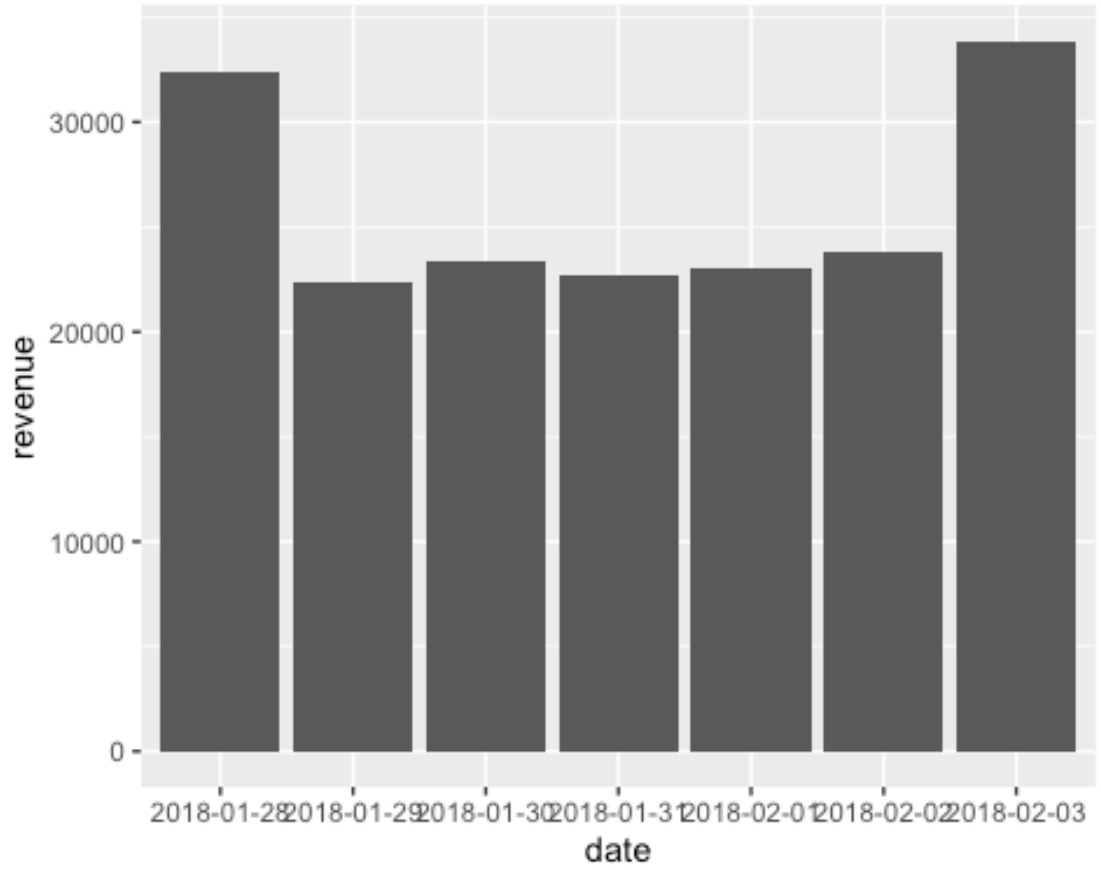
To do this, we can use the dplyr library.

```
# from sales1 df, we are selecting columns, date, store_id, revenue, then
grouping the column, first by date and then by store_id

EODTotalSales <- sales1 %>% select(date, store_id, revenue) %>%
group_by(date, store_id) %>% summarise("revenueSum" = sum(revenue))

## `summarise()` has grouped output by 'date'. You can override using the
## `.groups` argument.

EODTotalSales

## # A tibble: 864 × 3
## # Groups:   date [7]
##    date       store_id revenueSum
##    <chr>      <chr>         <dbl>
##  1 2018-01-28 S0001         237.
##  2 2018-01-28 S0002         420.
##  3 2018-01-28 S0003         176.
##  4 2018-01-28 S0004         127
##  5 2018-01-28 S0006          20.7
##  6 2018-01-28 S0008         195.
##  7 2018-01-28 S0009          55.3
##  8 2018-01-28 S0010         376.
##  9 2018-01-28 S0011          90.1
## 10 2018-01-28 S0012         150.
## # ℹ 854 more rows
```

### B) Is there a noted difference between the days?

In the data set, let's find whether there is any noted difference between the days.

```
revPerDay <- (sales1 %>% select(date, revenue) %>% group_by(date) %>%
summarise("revenue" = sum(revenue)))

ggplot(data = revPerDay, aes(x = date, y = revenue)) +
  geom_bar(position = "dodge", stat = "identity")
```




We can see on 28 Jan and 3 Feb, the revenue is more than 30,000 and for all other dates, they have almost similar revenue which is around 22500.

Let's check whether the difference is statistically significant.

```
table(sales1$date)

##
## 2018-01-28 2018-01-29 2018-01-30 2018-01-31 2018-02-01 2018-02-02 2018-02-
03
##      16639      16594      16617      16621      16710      16759
16717
```

As the sample size is large, we can assume its normally distributed.

let's do a one way test to see the statistical difference. For that we can **assume a significance level of 0.05**.

Let

- H0: There is no statistically significant difference between revenue of at least 2 different dates
- HA: There is statistically significant difference between revenue of at least 2 different dates

```
oneway.test(revenue ~ date,
            data = sales1)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  revenue and date
## F = 16.021, num df = 6, denom df = 51605, p-value < 2.2e-16
```

From the one way analysis, we have a **p-value of 2.2e-16** which is much **lesser than assumed significance level**. With this, we reject our null hypothesis and say that there **is statistically significant difference between revenue from at least two different dates.**

---

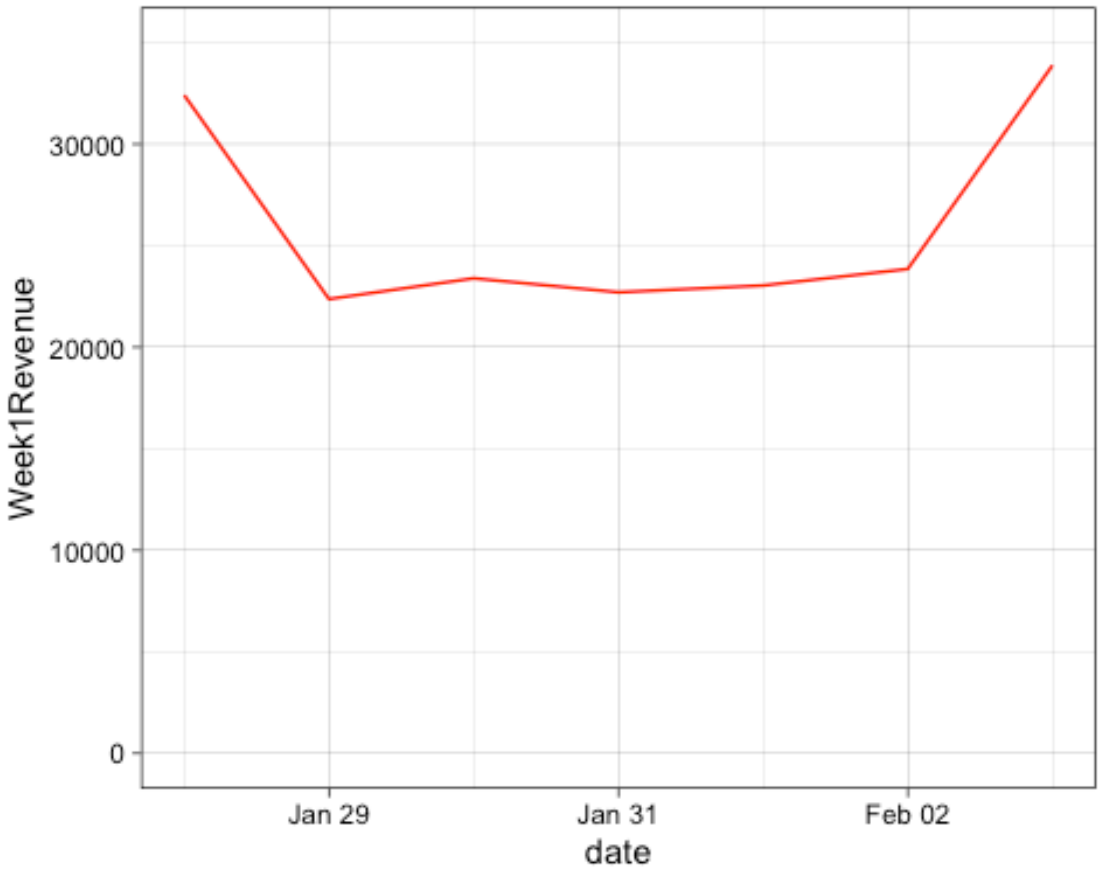
### C) Write also the code to calculate the total revenue over the seven day period. Plot it on a graph.

```
# for both datasets, using dplyr, select date and revenue, group the data by
date and find the sum of revenue. inorder to plot the date on graph, i
mutated the date column to change the data type of date column from character
to date.

(totalRevenue1 <- sales1 %>% select(date, revenue) %>%  group_by(date) %>%
summarise(Week1Revenue = sum(revenue)) %>% mutate(date, "date" =
as.Date(date, format = "%Y-%m-%d")))

## # A tibble: 7 × 2
##   date       Week1Revenue
##   <date>            <dbl>
## 1 2018-01-28       32409.
## 2 2018-01-29       22371.
## 3 2018-01-30       23393.
## 4 2018-01-31       22706.
## 5 2018-02-01       23039.
## 6 2018-02-02       23853.
## 7 2018-02-03       33881.

(totalRevenue2 <- sales2 %>% select(date, revenue) %>%  group_by(date) %>%
summarise(Week2Revenue = sum(revenue)) %>% mutate(date, "date" =
as.Date(date, format = "%Y-%m-%d")))
```

```
## # A tibble: 7 × 2
##   date       Week2Revenue
##   <date>            <dbl>
## 1 2018-04-28       33068.
## 2 2018-04-29       35264.
## 3 2018-04-30       28909.
## 4 2018-05-01       31619.
## 5 2018-05-02       33125.
## 6 2018-05-03       32711.
## 7 2018-05-04       33588.
```

From the table, we have information for total revenue for each day across all stores. But we can plot the data on a line plot so that its easier to interpret it. The below line plot shows trend for the first data set.

```
# using ggplot library, I took the dataset totalRevenue1, plotted date on x
axis and Week1Revenue on y axis.

ggplot(data = totalRevenue1, mapping = aes(x = date, y = Week1Revenue)) +
  geom_line(col = 'red') +
  ylim(0,35000) +
  theme_linedraw()
```



---

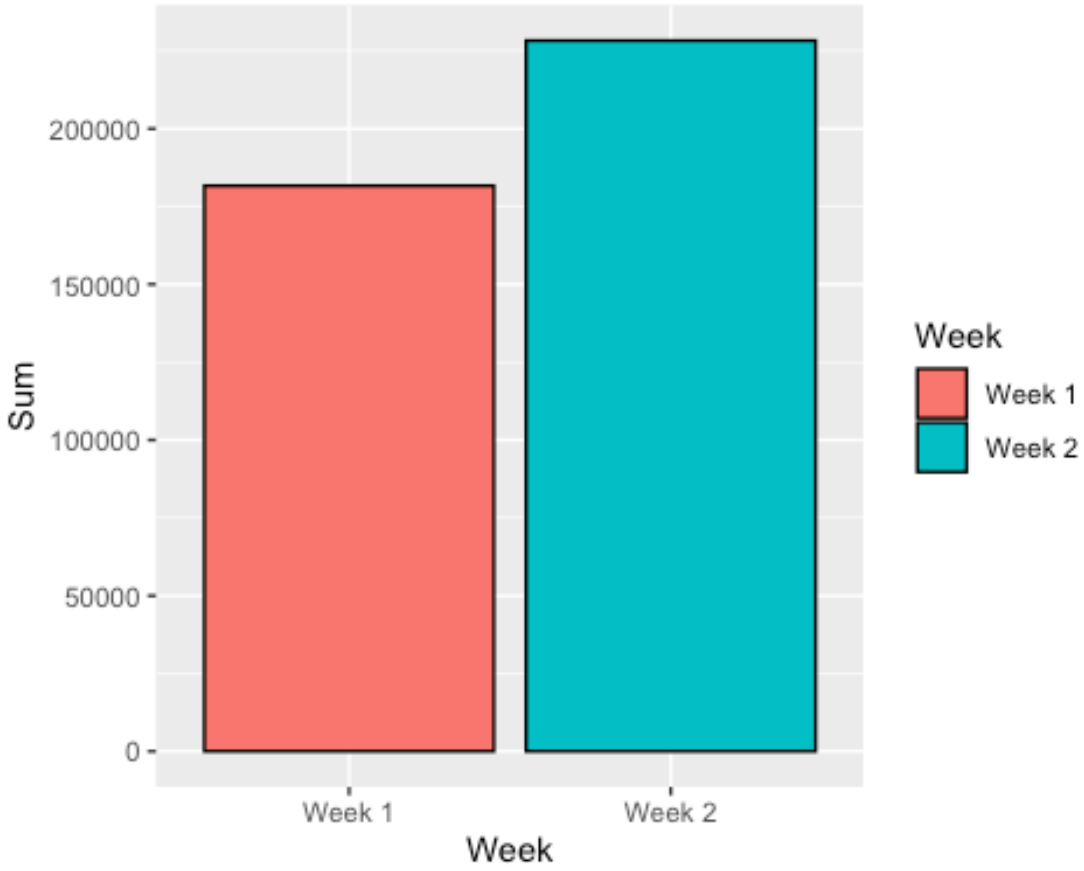
### D) Compare the revenue between the two data sets, is there a difference?

To compare the revenue from two data sets, we have to take sum of revenue of all dates.

```
# took total revenue from both datasets.

week1Sum <- sum(totalRevenue1$Week1Revenue)
week2Sum <- sum(totalRevenue2$Week2Revenue)

# using data.frame() function, I created two columns called Week and Sum.

weekSum <- data.frame(Week = c("Week 1", "Week 2"),
                      Sum = c(week1Sum, week2Sum))
```

Now that we have the required data frame, let's plot it on a bar graph.

```
ggplot(data = weekSum, aes(x = Week, y = Sum, fill = Week)) +
  geom_bar(stat='identity', position='dodge', col = 'black')
```



## Question 2

### What's the most popular product type (hierarchy 1) sold in all stores over a week?

### How much revenue did the stores receive for that product during the week?

### How does that compare with the second most popular product? Provide a table that shows the product type ranked from most to least popular.

### For each product type provide: how many subtypes (hierarchy 2) are there, how many products are in this product type, what's the sales quantity, and the revenue generated.

### Does this result vary between the two data sets?

To answer this question, we need the product_hierarchy data set. So we import it.

```
# import dataset using read.csv() function

heirarchy1 <- read.csv("Datasets/product_hierarchy.csv")
head(heirarchy1)
```

```
##   product_id product_length product_depth product_width cluster_id
## 1      P0000            5.0            20            12
## 2      P0001           13.5            22            20  cluster_5
## 3      P0002           22.0            40            22  cluster_0
## 4      P0004            2.0            13             4  cluster_3
## 5      P0005           16.0            30            16  cluster_9
## 6      P0006            8.5            15            15  cluster_0
##   hierarchy1_id hierarchy2_id hierarchy3_id hierarchy4_id hierarchy5_id
## 1           H00         H0004       H000401     H00040105   H0004010534
## 2           H01         H0105       H010501     H01050100   H0105010006
## 3           H03         H0315       H031508     H03150800   H0315080028
## 4           H03         H0314       H031405     H03140500   H0314050003
## 5           H03         H0312       H031211     H03121109   H0312110917
## 6           H03         H0316       H031608     H03160817   H0316081708
```

#### A) What's the most popular product type (hierarchy 1) sold in all stores over a week?

First, we will take required columns from sales1 data set.

```
hierarchySales <- sales1 %>% select(product_id, sales, revenue)
```

As we need hierarchy of products sold, we need to join the above data set with hierarchy data set.

```
# using inner_join(), join the hierarchySales and heirarchy1 data set with
product_id column as common.
```

```
head(productwithHierarchy <- hierarchySales %>% inner_join(heirarchy1, by =
"product_id"))
```

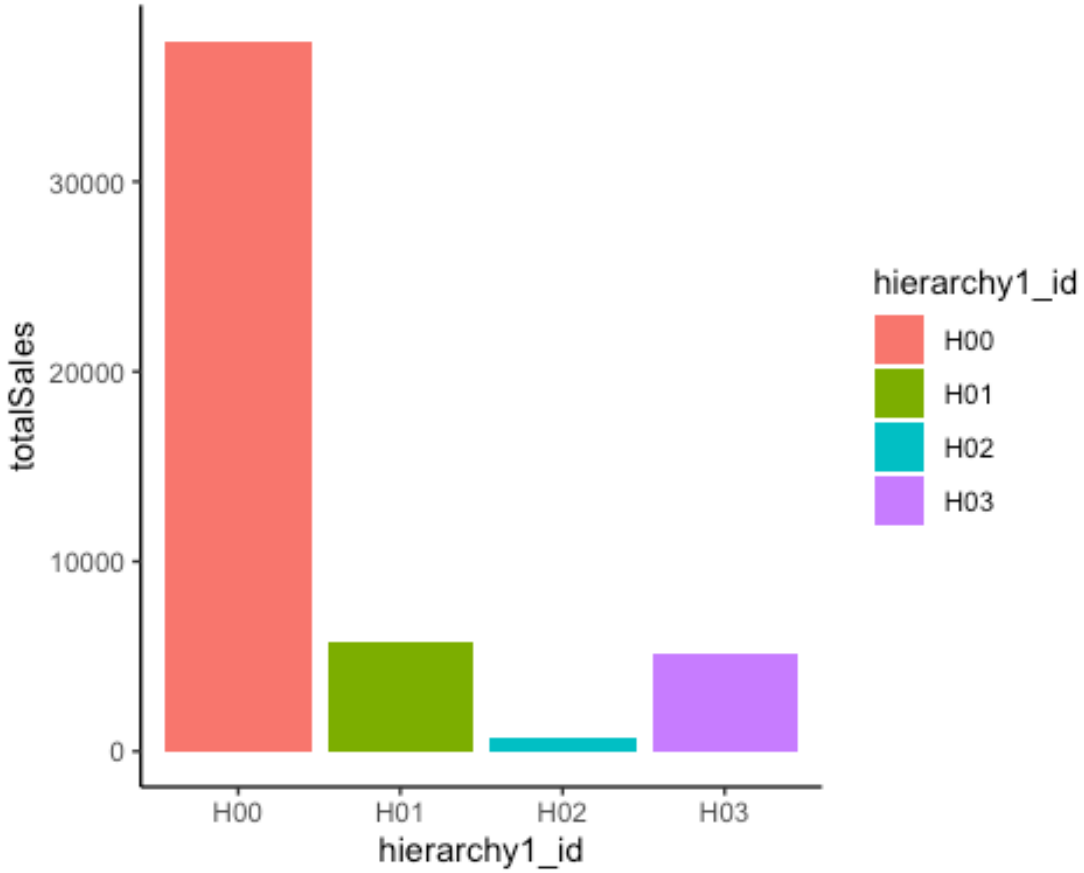
```
##   product_id sales revenue product_length product_depth product_width
## 1      P0001     0       0           13.5            22            20
## 2      P0001     0       0           13.5            22            20
## 3      P0001     0       0           13.5            22            20
## 4      P0001     0       0           13.5            22            20
## 5      P0001     0       0           13.5            22            20
## 6      P0001     0       0           13.5            22            20
##   cluster_id hierarchy1_id hierarchy2_id hierarchy3_id hierarchy4_id
## 1  cluster_5           H01         H0105       H010501     H01050100
## 2  cluster_5           H01         H0105       H010501     H01050100
## 3  cluster_5           H01         H0105       H010501     H01050100
## 4  cluster_5           H01         H0105       H010501     H01050100
## 5  cluster_5           H01         H0105       H010501     H01050100
## 6  cluster_5           H01         H0105       H010501     H01050100
##   hierarchy5_id
## 1   H0105010006
## 2   H0105010006
## 3   H0105010006
## 4   H0105010006
## 5   H0105010006
## 6   H0105010006
```

Using the 'productwithHierarchy' data set, group the hierarchy and we can find the total sales for each hierarchy, which represents the popularity of each hierarchy

```
# using group_by() function, group according to hierarchy and find total
sales for each hierarchy

(q2a <- productwithHierarchy %>% select(hierarchy1_id, sales) %>%
group_by(hierarchy1_id) %>% summarise(totalSales = sum(sales)))
```

```
## # A tibble: 4 × 2
##   hierarchy1_id totalSales
##   <chr>              <dbl>
## 1 H00               37429.
## 2 H01                5775
## 3 H02                 699.
## 4 H03                5129
```

```
ggplot(data = q2a, aes(x = hierarchy1_id, y = totalSales, fill =
hierarchy1_id)) +
  geom_bar(position = 'dodge', stat = 'identity') +
  theme_classic()
```



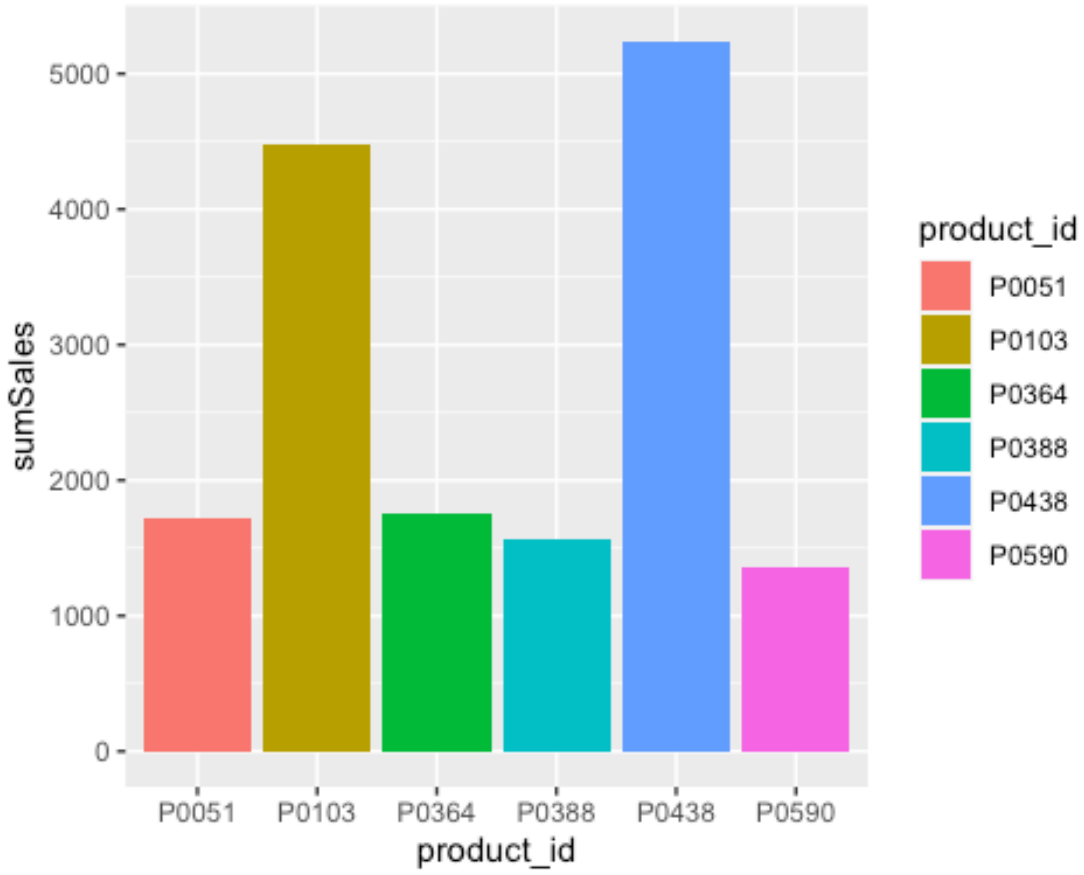

From the result, we can see H00 is the most popular product type, followed by H01.

Sales define how popular the product is. Therefore, to find the most popular product,

```
productSales <- sales1 %>% select(product_id, sales) %>% group_by(product_id)
%>% summarise("sumSales" = sum(sales)) %>% arrange(desc(sumSales))
head(productSales)
```

```
## # A tibble: 6 × 2
##   product_id sumSales
##   <chr>         <dbl>
## 1 P0438          5245
## 2 P0103          4485
## 3 P0364          1748
## 4 P0051          1728
## 5 P0388          1571
## 6 P0590          1360
```

```
ggplot(data = head(productSales), aes(x = product_id, y = sumSales, fill =
product_id)) +
  geom_bar(position = 'dodge', stat = 'identity')
```




**P0438 is the most popular product, followed by P0438.**

---

### B) How much revenue did the stores receive for that product during the week?

We can calculate the revenue stores received from H00 type product.

```
# using filter, select H00 and find sum the revenue

productwithHierarchy %>% filter(hierarchy1_id == "H00") %>%
group_by(hierarchy1_id) %>% summarise(sum(revenue))
```

```
## # A tibble: 1 × 2
##   hierarchy1_id `sum(revenue)`
##   <chr>                  <dbl>
## 1 H00                   95054.
```

We can see the **revenue of product type H00 is 95053.8**

Now, to find the revenue of the popular product,

```
(product1Revn <- sales1 %>% filter(product_id == "P0438") %>%
select(store_id, revenue) %>% summarise(sum(revenue)))
```

```
##   sum(revenue)
## 1      2428.19
```

The **product revenue is 2428.19**

---

### C) How does that compare with the second most popular product?

The second most popular product from product Sales data set is P0103.

```
# using filter, select H00 and find sum the revenue

productwithHierarchy %>% filter(hierarchy1_id == "H01") %>%
group_by(hierarchy1_id) %>% summarise("revenue" = sum(revenue))
```

```
## # A tibble: 1 × 2
##   hierarchy1_id revenue
##   <chr>           <dbl>
## 1 H01            58558.
```

We can see the **revenue of product type H01 is 58558.12**

Now, to find the revenue of the popular product,

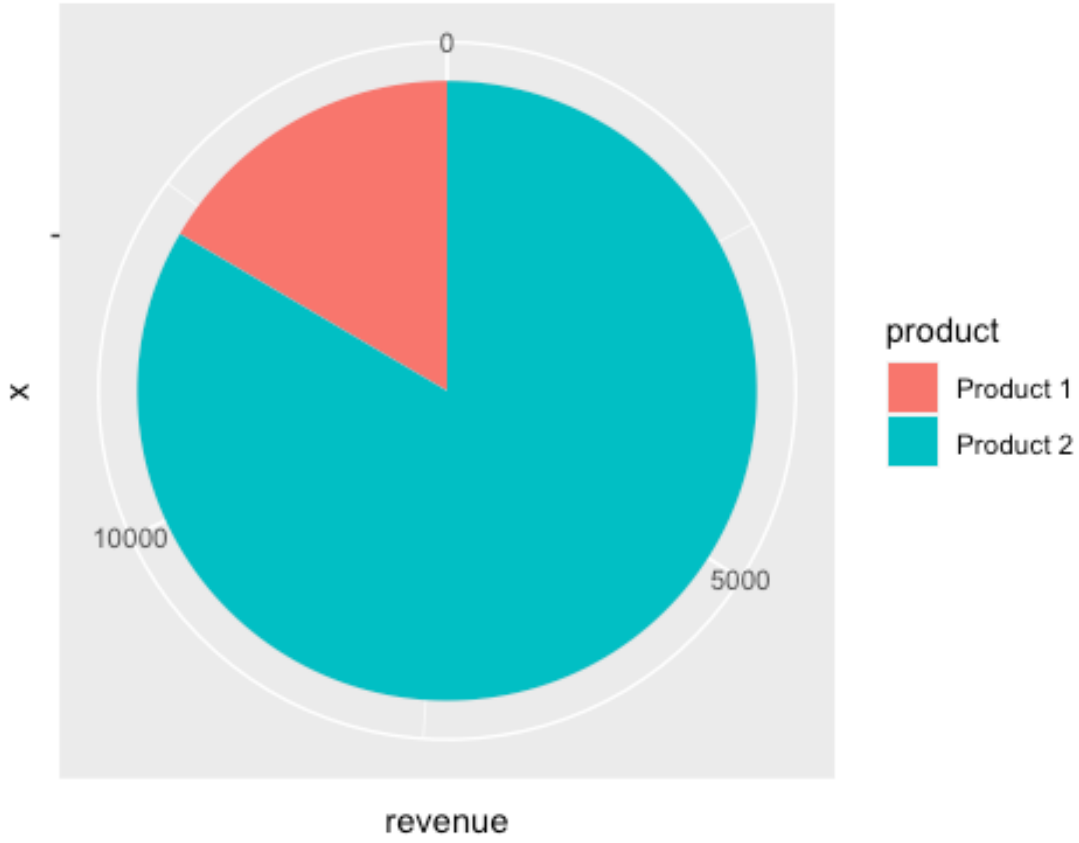
```
(product2Revn <- sales1 %>% filter(product_id == "P0103") %>%
select(store_id, revenue) %>% summarise("revenue" = sum(revenue)))
```

```
##    revenue
## 1 12250.26
```

The **product revenue is 12250.26**

Now let's plot the most popular two products

```
prodDf <- data.frame(product = c("Product 1", "Product 2"),
                     revenue = c(product1Revn$`sum(revenue)`,
product2Revn$revenue))

ggplot(data = prodDf, aes(x = "", y = revenue, fill = product)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0)
```




---

**D) Provide a table that shows the product type ranked from most to least popular**

```
rank <- sales1 %>% select(product_id, sales) %>% inner_join(heirarchy1, by = "product_id")

rank %>% group_by(hierarchy1_id) %>% summarise("totalSales" = sum(sales)) %>% arrange(desc(totalSales))
```

```
## # A tibble: 4 × 2
##   hierarchy1_id totalSales
##   <chr>              <dbl>
## 1 H00               37429.
## 2 H01                5775
## 3 H03                5129
## 4 H02                 699.
```

---

**E) For each product type provide: how many subtypes (hierarchy 2) are there, how many products are in this product type, what's the sales quantity, and the revenue generated.**

**how many subtypes (hierarchy 2) are there ?**

To find the number of sub types of each hierarchy, we have to count after grouping each hierarchy.

```
# from hierarchy1, group it by hierarchy1_id, count the hierarchy2_id and
then count the hierarchy1_id

(subtypeCount <- heirarchy1 %>% group_by(hierarchy1_id) %>%
count(hierarchy2_id) %>% count(hierarchy1_id))
```

```
## # A tibble: 4 × 2
## # Groups:   hierarchy1_id [4]
##   hierarchy1_id     n
##   <chr>         <int>
## 1 H00               5
## 2 H01               4
## 3 H02               2
## 4 H03               7
```

The above response shows number of sub types in each hierarchy.

**how many products are in this product type ?**

using hierarchy1 data set, we can count how many times hierarchy1_id is mentioned as for every product, the former is mentioned in the same row.

```
# group hierarchy1_id and then count it.

heirarchy1 %>% group_by(hierarchy1_id) %>% count(hierarchy1_id)
```

```
## # A tibble: 4 × 2
## # Groups:   hierarchy1_id [4]
##   hierarchy1_id     n
##   <chr>         <int>
## 1 H00             215
## 2 H01             181
## 3 H02              11
## 4 H03             292
```

The above response shows the number of products in each hierarchy.

**what's the sales quantity and revenue generated ?**

Let's find the sales for each hierarchy. For that we have to join the sales data set for both weeks and the hierarchy data set.

```
# unse the innerjoin function to join sales and hierarchy dataset.

byTypeSales1 <- sales1 %>% select(product_id, sales, revenue) %>%
inner_join(heirarchy1, by = "product_id")

byTypeSales2 <- sales2 %>% select(product_id, sales, revenue) %>%
inner_join(heirarchy1, by = "product_id")
```

Let's find for first data set.

- **Week 1**

```
# using the above created table, group it by hierarchy1_id and sum the sales
value.

(sales1PerType <- byTypeSales1 %>% select(hierarchy1_id, sales, revenue) %>%
group_by(hierarchy1_id) %>% summarise(sum(sales)))
```

```
## # A tibble: 4 × 2
##   hierarchy1_id `sum(sales)`
##   <chr>               <dbl>
## 1 H00                37429.
## 2 H01                 5775
## 3 H02                  699.
## 4 H03                 5129
```

Do the same for second data set.

- **Week 2**

```
# using the above created table, group it by hierarchy1_id and sum the sales
value.

(sales2PerType <- byTypeSales2 %>% select(hierarchy1_id, sales, revenue) %>%
group_by(hierarchy1_id) %>% summarise(sum(sales)))
```

```
## # A tibble: 4 × 2
##   hierarchy1_id `sum(sales)`
##   <chr>               <dbl>
## 1 H00                40879.
## 2 H01                 7728
## 3 H02                   83
## 4 H03                 7473
```

Let's now find the Revenue generated by each hierarchy1_id.

```
# use the innerjoin function to join sales and hierarchy dataset.

(revenue1PerType <- byTypeSales1 %>% select(hierarchy1_id, sales, revenue)
%>% group_by(hierarchy1_id) %>% summarise(sum(revenue)))
```

```
## # A tibble: 4 × 2
##   hierarchy1_id `sum(revenue)`
##   <chr>                 <dbl>
## 1 H00                  95054.
## 2 H01                  58558.
## 3 H02                   3540.
## 4 H03                  24500.
```

```
# using the above created table, group it by hierarchy1_id and sum the
revenue value.
```

```
(revenue2PerType <- byTypeSales2 %>% select(hierarchy1_id, sales, revenue)
%>% group_by(hierarchy1_id) %>% summarise(sum(revenue)))
```

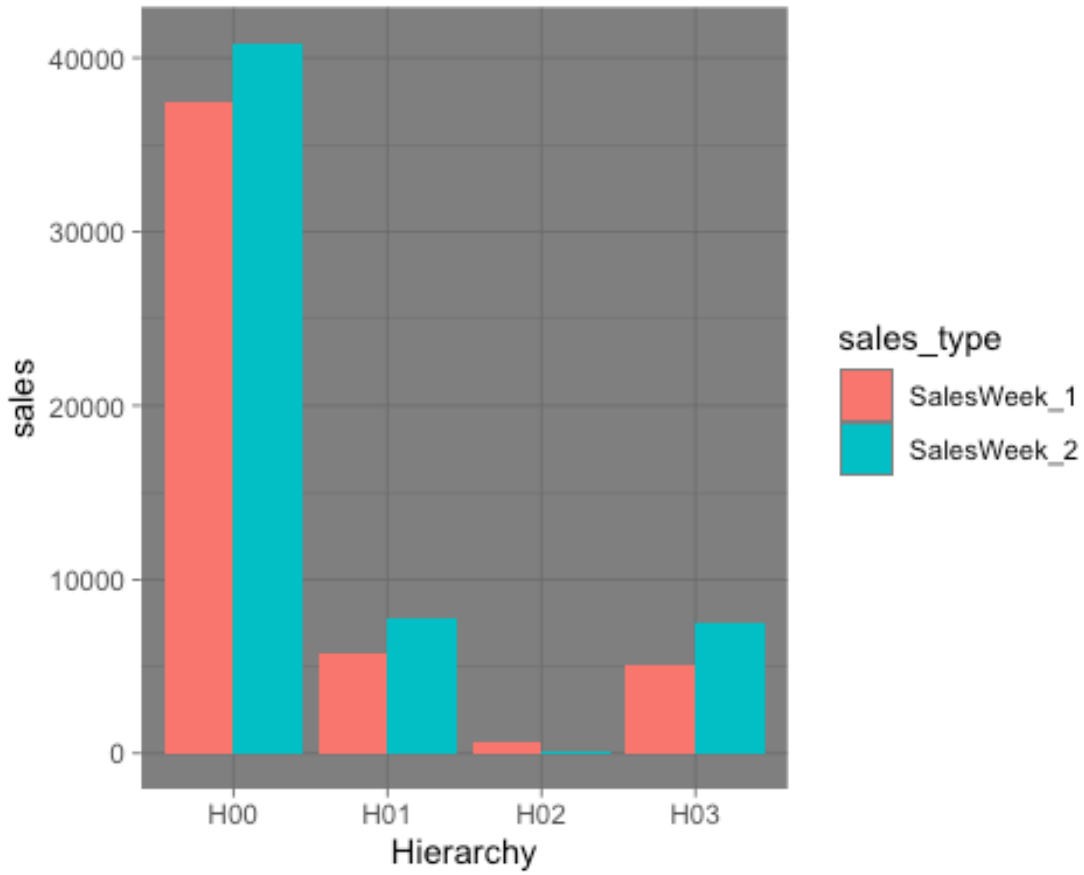
```
## # A tibble: 4 × 2
##   hierarchy1_id `sum(revenue)`
##   <chr>                  <dbl>
## 1 H00                  112542.
## 2 H01                   77139.
## 3 H02                     180.
## 4 H03                   38425.
```

provide visualization below.

```
srdf <- data.frame("Hierarchy" = sales1PerType$hierarchy1_id,
                   "SalesWeek_1" = sales1PerType$`sum(sales)`,
                   "SalesWeek_2" = sales2PerType$`sum(sales)`)
```

using the above create data frame, let's create a side by side bar plot to visualize the information.

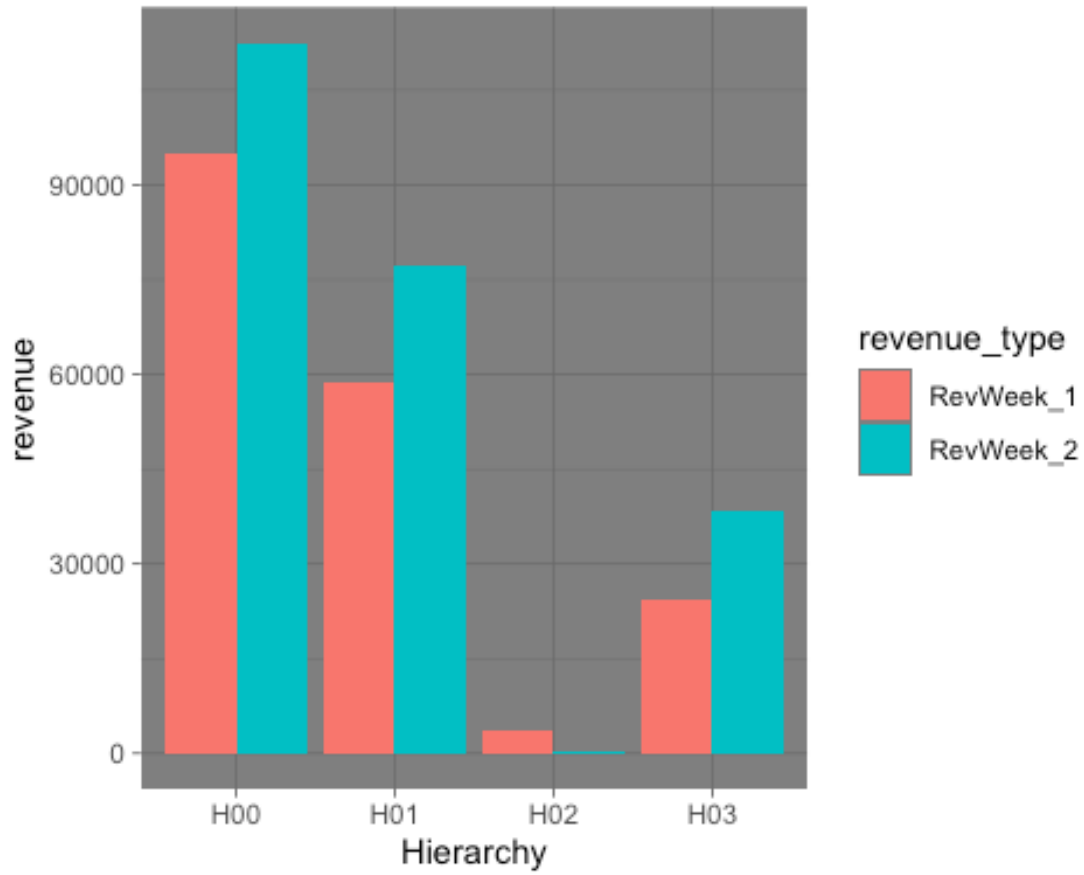
```
# gather() to reshape the data into a longer format. It specifies that we
want to gather the sales1 and sales2 columns into a new column called
sales_type and their corresponding values into a new column called sales. The
-product part indicates that we want to exclude the product column from being
gathered.

ggplot(gather(srdf, sales_type, sales, -Hierarchy),
       aes(x = Hierarchy, y = sales, fill = sales_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_dark()
```




From the above graph, we can easily visualise the sales info between two data sets.

Let's do the same with the revenue information.

```
rdf <- data.frame("Hierarchy" = revenue1PerType$hierarchy1_id,
                  "RevWeek_1" = revenue1PerType$`sum(revenue)`,
                  "RevWeek_2" = revenue2PerType$`sum(revenue)`)

# gather() to reshape the data into a longer format. It specifies that we
want to gather the sales1 and sales2 columns into a new column called
sales_type and their corresponding values into a new column called sales. The
-product part indicates that we want to exclude the product column from being
gathered.

ggplot(gather(rdf, revenue_type, revenue, -Hierarchy),
       aes(x = Hierarchy, y = revenue, fill = revenue_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_dark()
```




The above graph gives us comparison between two data sets.

## Question 3

### Compare the sales volumes between the two most common store types in the data set.

### How do they compare in terms of total revenue? Is there a relationship between a store's size and its revenue?

### Looking at the revenues between all the store types, what other factors could affect the sales numbers and revenue?

### Write the code to verify your hypothesis.

```
storeData <- read.csv("Datasets/store_cities.csv")
head(storeData)
```

```
##   store_id storetype_id store_size city_id
## 1    S0091         ST04         19    C013
## 2    S0012         ST04         28    C005
```

```
## 3    S0045         ST04         17    C008
## 4    S0032         ST03         14    C019
## 5    S0027         ST04         24    C022
## 6    S0088         ST04         20    C009
```

### A) Compare the sales volumes between the two most common store types in the data set.

To find that, we have to merge store data with the sales data set.

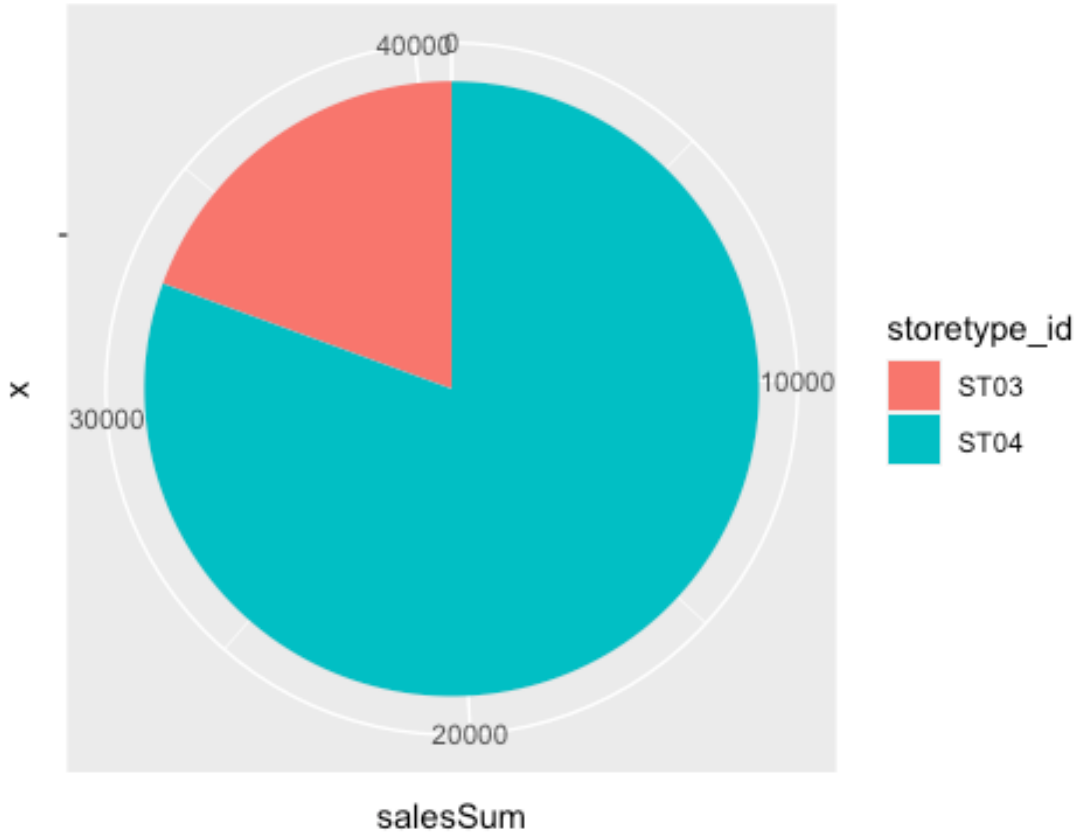
```
# create a table by selecting store_id from sales.

storeCount <- sales1 %>% select(store_id)

# using the above table, inner join it with store Data, find the common store
type.

(storeTypeCount <- storeCount %>% inner_join(storeData, by = "store_id") %>%
count(storetype_id) %>% arrange(desc(n)))

##   storetype_id     n
## 1         ST04 86496
## 2         ST03 18716
## 3         ST01  7669
## 4         ST02  3776
```

From the above table, we can see ST04 and ST03 are two most common store types.

**Compare sales volumes**

Let's compare the sales volumes between these two store types.

```
# join storeData with sales data.

storeTypeSalesRevenue <- sales1 %>% select(store_id, sales, revenue) %>%
inner_join(storeData, by = "store_id")

# using filter, filter out sales volume of store which are common

compareSalesVol <- storeTypeSalesRevenue %>% filter(storetype_id == "ST03" |
storetype_id == "ST04") %>% group_by(storetype_id) %>% summarise("salesSum" =
sum(sales))

# using the above table, create a pie chart to represent the proportion of
sale between the those store types.

ggplot(data = compareSalesVol, aes(x = " ", y = salesSum, fill =
storetype_id)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(start = 0, "y")
```

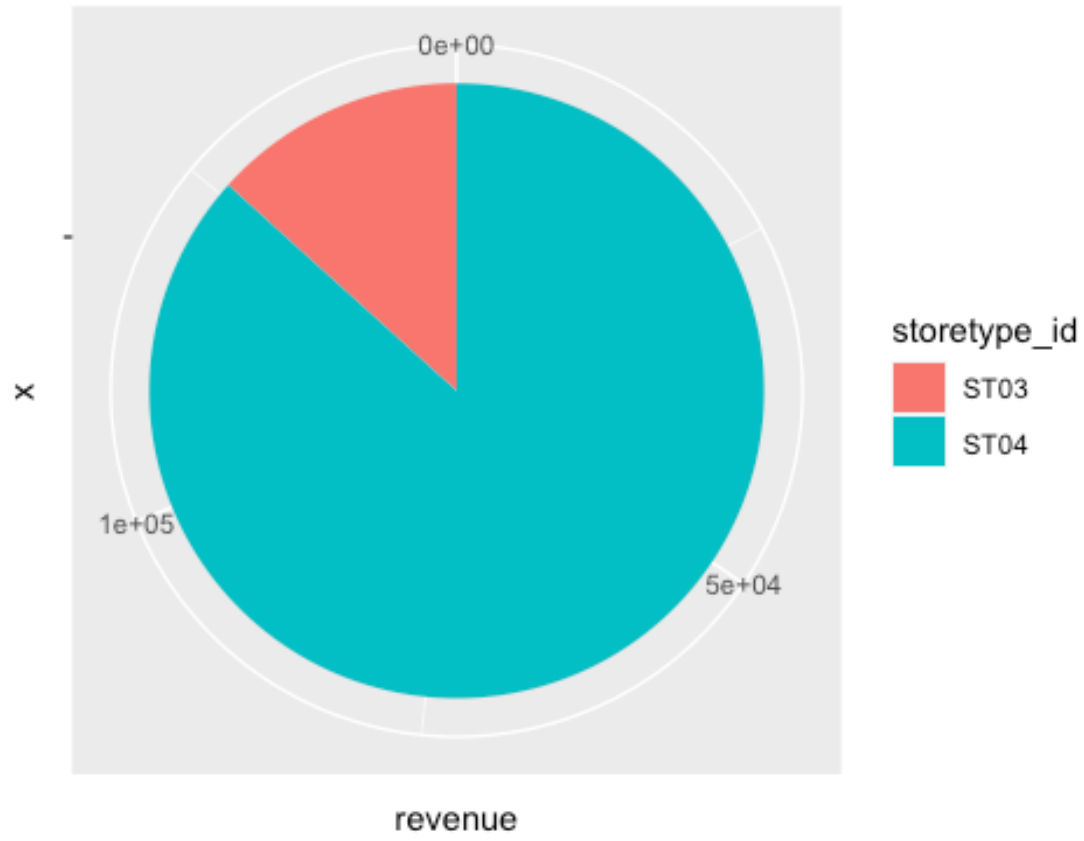



---

### B) How do they compare in terms of total revenue?

Let's find the revenue total just like we have did above.

```
strRev <- storeTypeSalesRevenue %>% filter(storetype_id == "ST03" |
storetype_id == "ST04") %>% group_by(storetype_id) %>% summarise("revenue" =
sum(revenue))

# using the above table, create a pie chart to represent the proportion of
revenue between the those store types.

ggplot(data = strRev, aes(x = " ", y = revenue, fill = storetype_id)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(start = 0, "y")
```




The revenue from ST04 is much higher compared to ST03. This can we because the sales of ST04 is also significantly higher than the other.

---

### C) Is there a relationship between a store's size and its revenue?

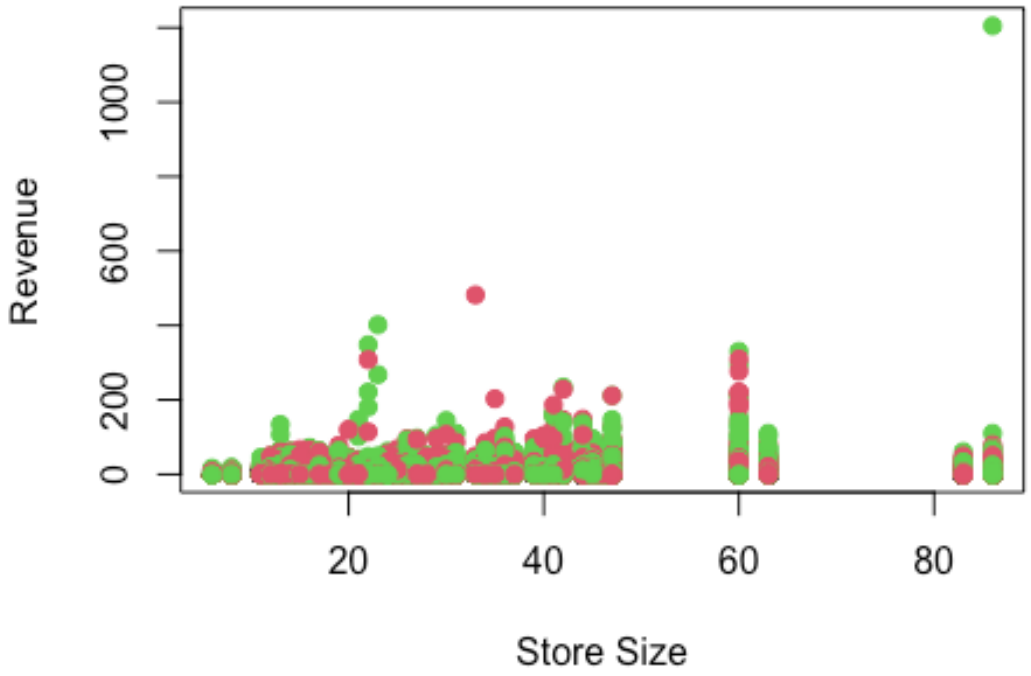
To find the the relationship between store size and revenue, let's find the the correlation value between them.

```
# find the correlation which gives us the strength of relationship between
the comparing variables.

cor(x = storeTypeSalesRevenue$store_size, y = storeTypeSalesRevenue$revenue)

## [1] 0.07149428
```

Also, let's check visually whether there is any trend using a scatter plot.

```
# create a scatterplot to visually understand whether there is any trend
among the

plot(x = storeTypeSalesRevenue$store_size, y = storeTypeSalesRevenue$revenue,
     pch = 19,
```

```
     xlab = "Store Size",
     ylab = "Revenue",
     col = 2:3)
```




**From the correlation value, and the scatter plot, it is clear that there is no any kind of relation between the store size and revenue.**

---

**D) Looking at the revenues between all the store types, what other factors could affect the sales numbers and revenue? Write the code to verify your hypothesis.**

1. **number of products and type of product affects the revenue of these stores.**

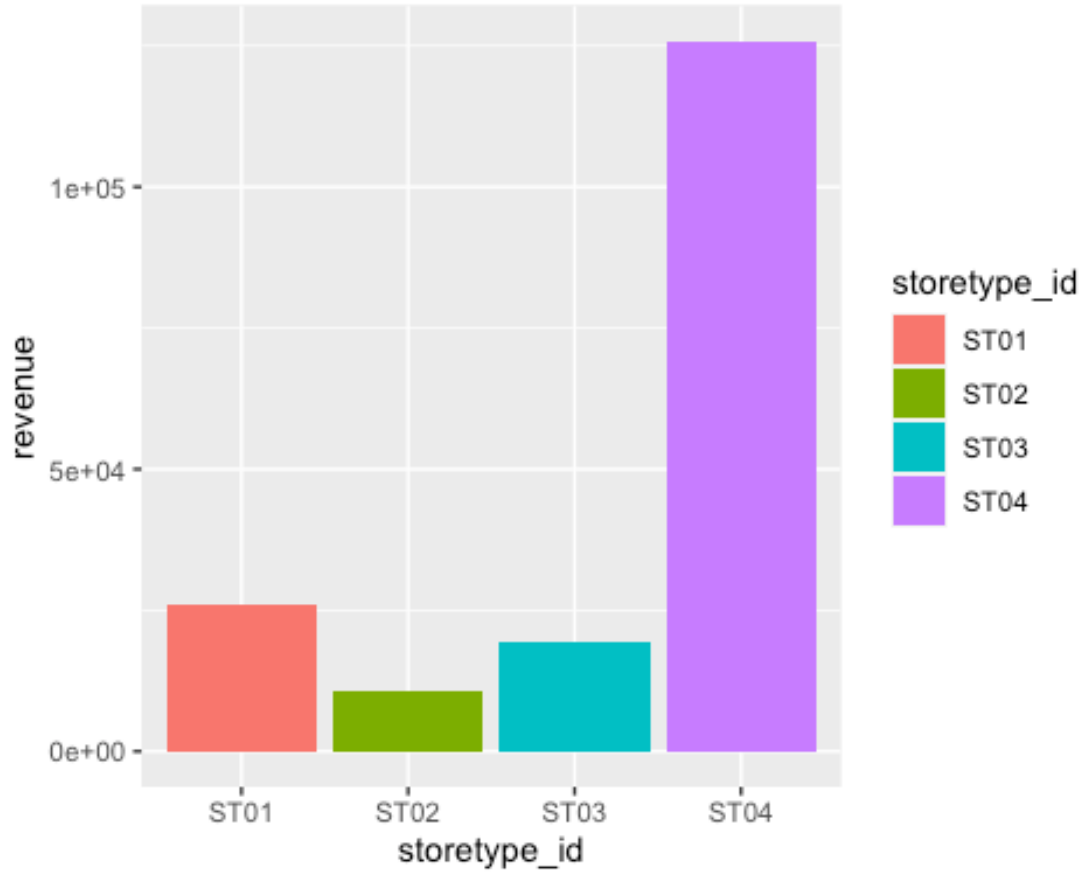
For that, we have to join sales, hierarchy and store Data.

```
subSales <- sales1 %>% select(product_id, store_id, sales, revenue)
subHierarchy <- heirarchy1 %>% select(product_id, hierarchy1_id,
hierarchy2_id)

subSalesHierarchy <- subSales %>% inner_join(subHierarchy, by = "product_id")

joinedAll <- subSalesHierarchy %>% inner_join(storeData, by = "store_id")
head(joinedAll)
```

```
##   product_id store_id sales revenue hierarchy1_id hierarchy2_id
storetype_id
## 1      P0001    S0001     0       0           H01         H0105
ST04
## 2      P0001    S0002     0       0           H01         H0105
ST04
## 3      P0001    S0004     0       0           H01         H0105
ST04
## 4      P0001    S0008     0       0           H01         H0105
ST04
## 5      P0001    S0012     0       0           H01         H0105
ST04
## 6      P0001    S0013     0       0           H01         H0105
ST04
##   store_size city_id
## 1         41    C031
## 2         39    C007
## 3         20    C022
## 4         27    C024
## 5         28    C005
## 6         33    C026
```

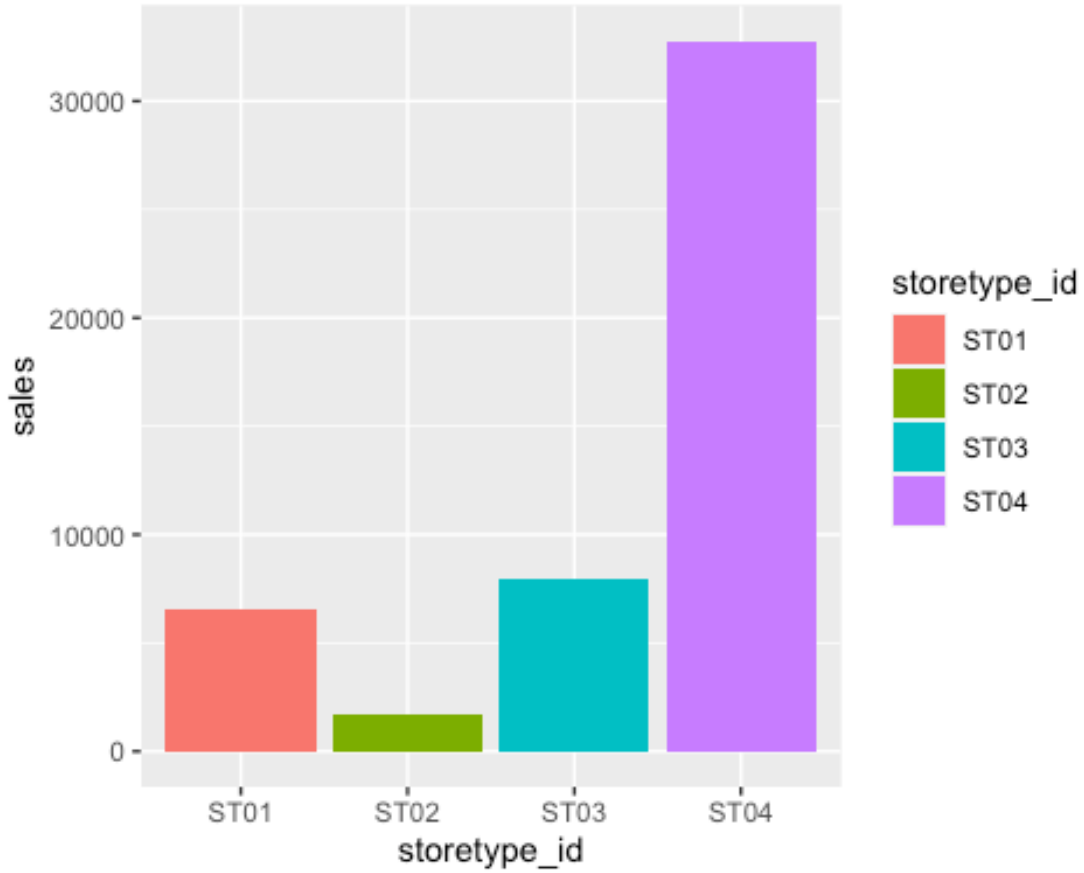
Looking at the revenue,

```
typeRev <- joinedAll %>% select(storetype_id, revenue, sales) %>%
group_by(storetype_id) %>% summarise("revenue" = sum(revenue))

ggplot(data = typeRev, aes(x = storetype_id, y = revenue, fill =
storetype_id)) +
  geom_bar(stat = "identity", position = "dodge")
```




Looking at the sales volume,

```
typeSales <- joinedAll %>% select(storetype_id, sales) %>%
group_by(storetype_id) %>% summarise("sales" = sum(sales))

ggplot(data = typeSales, aes(x = storetype_id, y = sales, fill =
storetype_id)) +
  geom_bar(stat = "identity", position = "dodge")
```




**The store type also affects the sales and revenue as shown in the above two graphs.**

## Question 4

### Several different types of promotions were applied to the products during the period with various level of promotion rates.

### Pick one of the data sets, for each promotion type, display the different levels of promotion used during the period.

### Analyse the effectiveness of the promotion on the sales of the products.

### Compare the results between the two time periods.

#### A) Pick one of the data sets, for each promotion type, display the different levels of promotion used during the period.

```
head(sales1 %>% select(promo_type_1, promo_bin_1) %>% distinct(promo_type_1,
promo_bin_1), 10)
```

```
##    promo_type_1 promo_bin_1
## 1           PR14
```

```
## 2          PR10     verylow
## 3          PR03     verylow
## 4          PR05    moderate
## 5          PR05        high
## 6          PR09         low
## 7          PR10         low
## 8          PR05         low
## 9          PR06    moderate
## 10         PR09        high
```

Above table shows the promo type and its related promo bin.

---

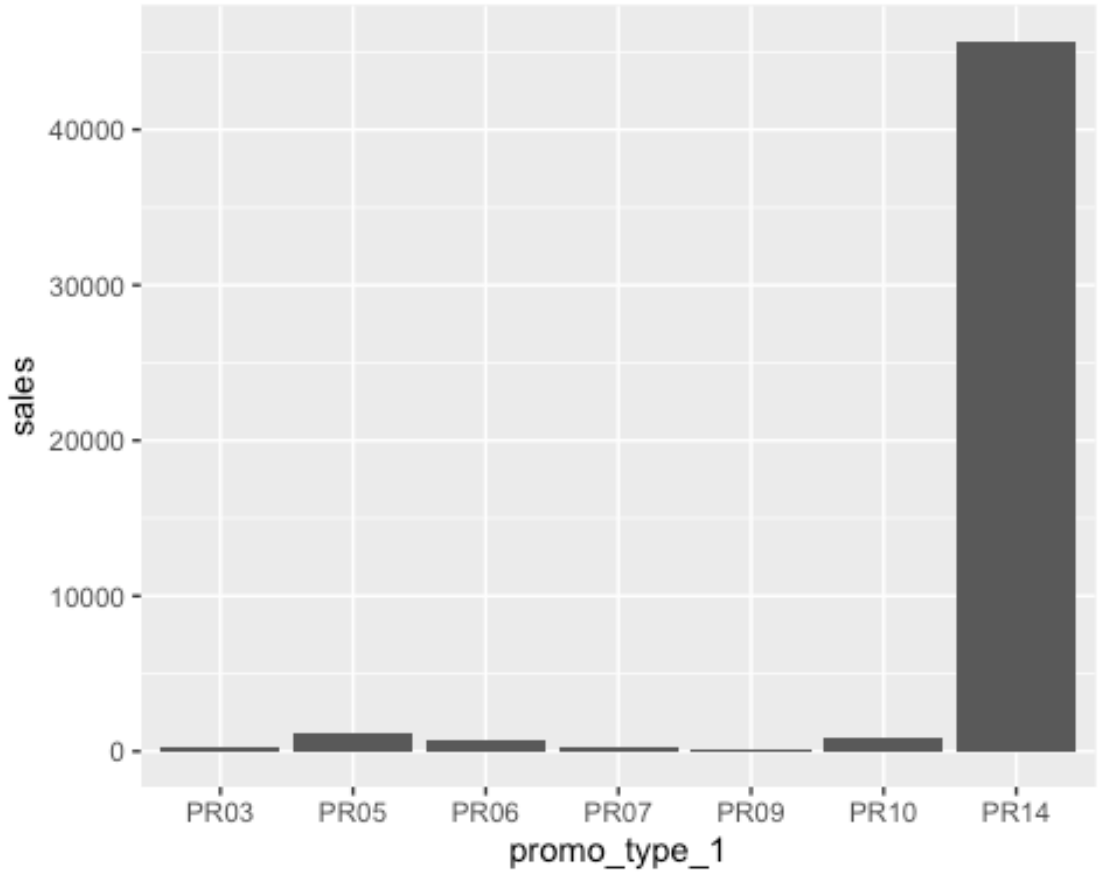
### B) Analyse the effectiveness of the promotion on the sales of the products.

```
(salesPerPromo <- sales1 %>% select(promo_type_1, sales) %>%
group_by(promo_type_1) %>% summarise("sales" = sum(sales)))
```

```
## # A tibble: 7 × 2
##   promo_type_1  sales
##   <chr>         <dbl>
## 1 PR03            258
## 2 PR05           1144
## 3 PR06            653
## 4 PR07            312
## 5 PR09            123
## 6 PR10            810
## 7 PR14          45732.
```

Let's plot a graph to understand the result effectively,

```
ggplot(data = salesPerPromo, aes(x = promo_type_1, y = sales)) +
  geom_bar(stat = "identity", position = "dodge")
```




The promo type PR14 has performed significantly higher than other promo types.

---

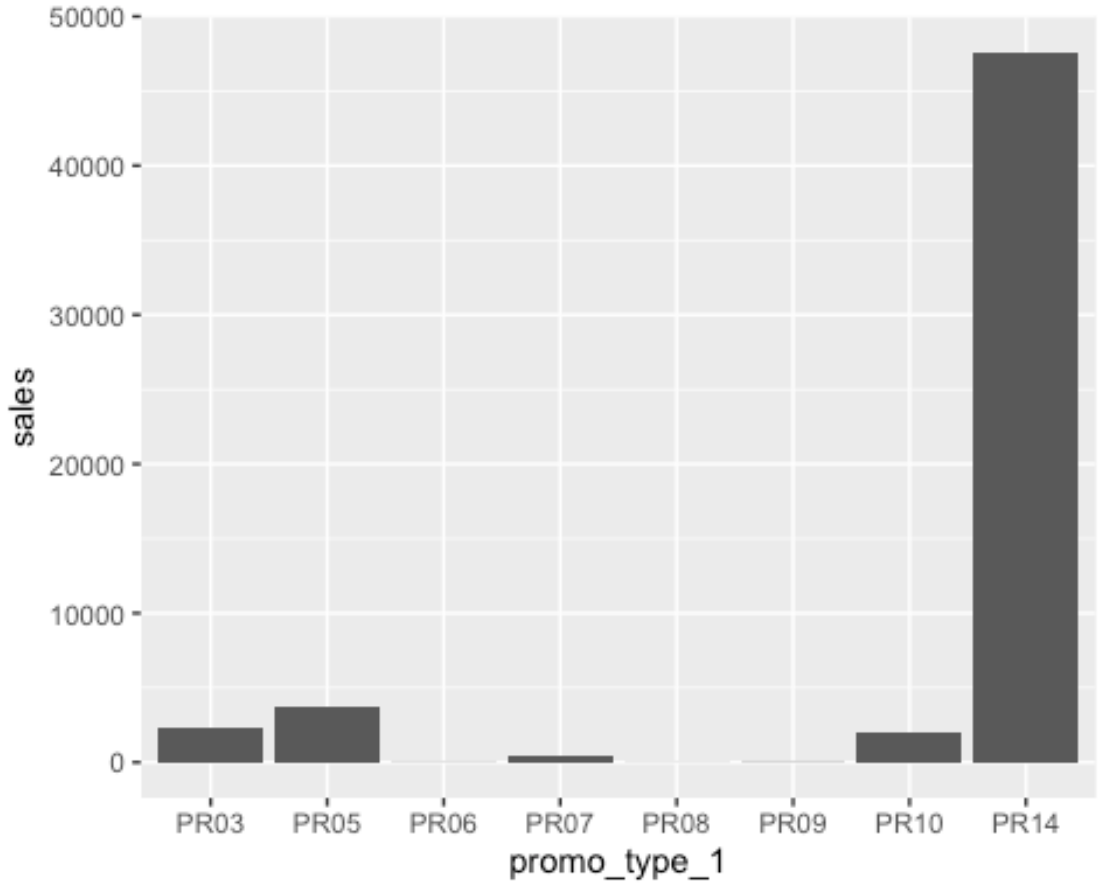
### C) Compare the results between the two time periods.

Doing the same analysis on the second data set below,

```
(sales2PerPromo <- sales2 %>% select(promo_type_1, sales) %>%
group_by(promo_type_1) %>% summarise("sales" = sum(sales)))
```

```
## # A tibble: 8 × 2
##   promo_type_1  sales
##   <chr>         <dbl>
## 1 PR03           2290
## 2 PR05           3763
## 3 PR06             26
## 4 PR07            374
## 5 PR08             18
## 6 PR09             38
## 7 PR10           2011
## 8 PR14          47643.
```

```
ggplot(data = sales2PerPromo, aes(x = promo_type_1, y = sales)) +
  geom_bar(stat = "identity", position = "dodge")
```




In the second data set, again we can see The promo type PR14 performance is significantly higher than other types. Also, in the second data set, promo types like PR03, PR04, PR10 also performs better compared to the first data set.