ADanalysisProgram

Import the libraries we need

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------------- tidyverse 
```

```
## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.0     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ---------------------------------------------------------------------- tidyverse_confli
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(readr)
library(stringr)
library(dplyr)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

Batch import the csv files in folders

```

path <- "/Users/yuanpili/OneDrive - Technological University Dublin/2020-2021 year 3/statistical progra
files <- list.files(path=path, pattern="*.csv")
for(file in files)
{
  perpos <- which(strsplit(file, "")[[1]]==".")
  assign(
    gsub(" ","",substr(file, 1, perpos-1)),
    read.csv(paste(path,file,sep="")))
}
```

Change the path name as your own file path

```
## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## incomplete final line found by readTableHeader on '/Users/yuanpili/OneDrive
## - Technological University Dublin/2020-2021 year 3/statistical programming/
## dataset/advertiser.csv'
```

In the environment on the right shows that all the CSV files have been imported

Validating primary keys

```
advertiser_id_counts <- count(advertiser, ID)
filter(advertiser_id_counts, n>1)
```

Validate that primary keys of 'advertiser' and 'campaigns' are unique. There should be no counts greater than 1.

```
## [1] ID n
## <0 rows> (or 0-length row.names)
```

```
campaigns_id_counts <- count(campaigns, id)
filter(campaigns_id_counts, n>1)
```

```
## [1] id n
## <0 rows> (or 0-length row.names)
```

Join all the tables together into 'res' that contains advertiser__id, advertiser__name, campaign__id, campaign__name, budget, clicks(number of clicks), impressions(number of impressions), conversions(number of conversions)

```
click <- count(clicks, campaign_id)
impression <- count(impressions, campaign_id)
conversion <- count(conversions, campaign_id)

# since we need to change the variable name after each join, so pipe doesn't work here the best
res <- full_join(advertiser,campaigns,by=c('ID'='advertiser_id'))
names(res)[names(res)=="id"]="campaign_id"
names(res)[names(res)=='ID']='advertiser_id'
names(res)[names(res)=='name.x']='advertiser_name'
names(res)[names(res)=='name.y']='campaign_name'

res <- full_join(res,click,by=c('campaign_id'='campaign_id'))
names(res)[names(res)=="n"]="clicks"
res$clicks[which(is.na(res$clicks))]<-0

res <- full_join(res,impression,by=c('campaign_id'='campaign_id'))
names(res)[names(res)=='n']='impressions'
```

```
res <- full_join(res,conversion,by=c('campaign_id'='campaign_id'))
names(res)[names(res)=='n']='conversions'
res$conversions[which(is.na(res$conversions))]<-0

res <- na.omit(res) # delete the rows(impressions) that contain NA value
```

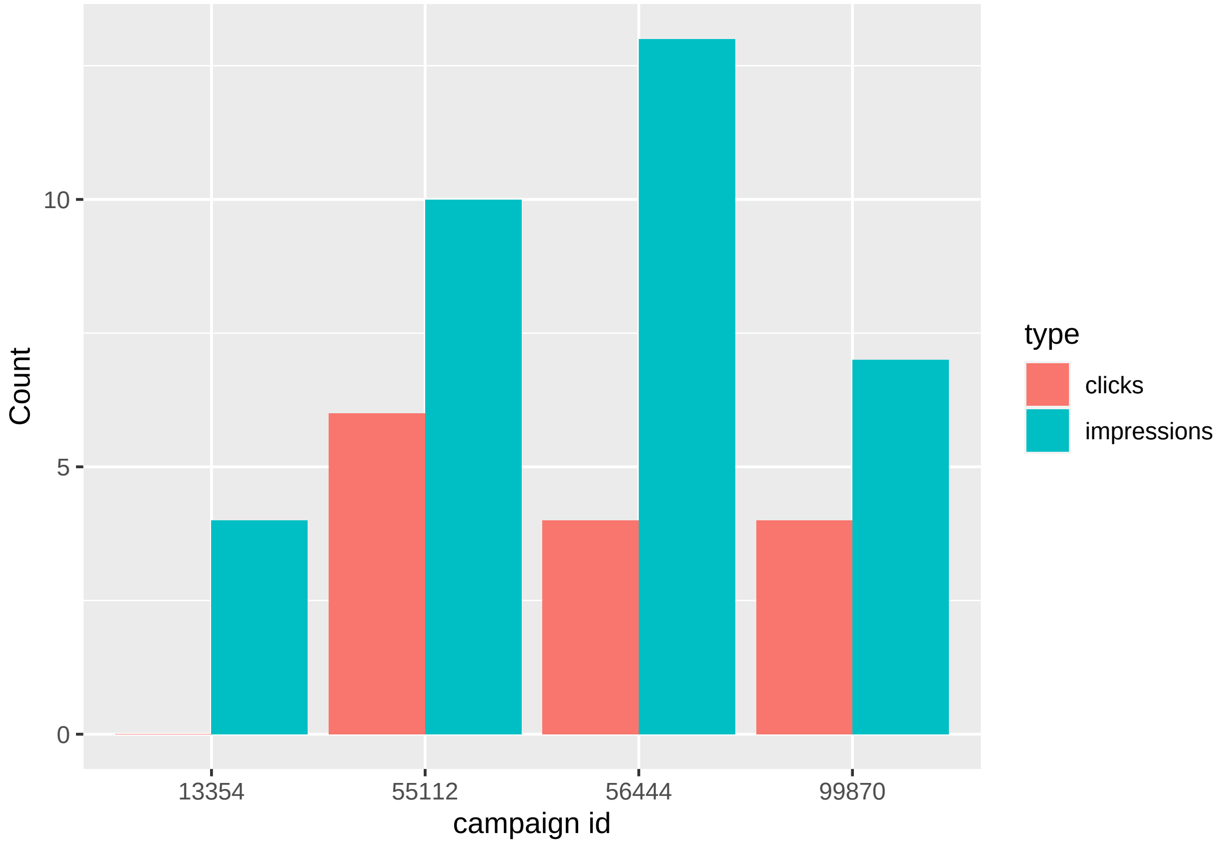
Analysis CTR, CPC, CPM, conversion_rate

```
res <- mutate(res, CTR = impressions / clicks) %>%  # Click-through rate = impressions / clicks
  mutate(CPC = budget / clicks) %>% # Cost per click (CPC) = total cost/number of clicks
  mutate(CPM = (budget / impressions) * 1000) %>% # Cost Per Thousand Impression (CPM) = (budget/number
  mutate(conversion_rate = conversions / clicks)
```

Graph of the numbers of clicks and impressions vs campaign id

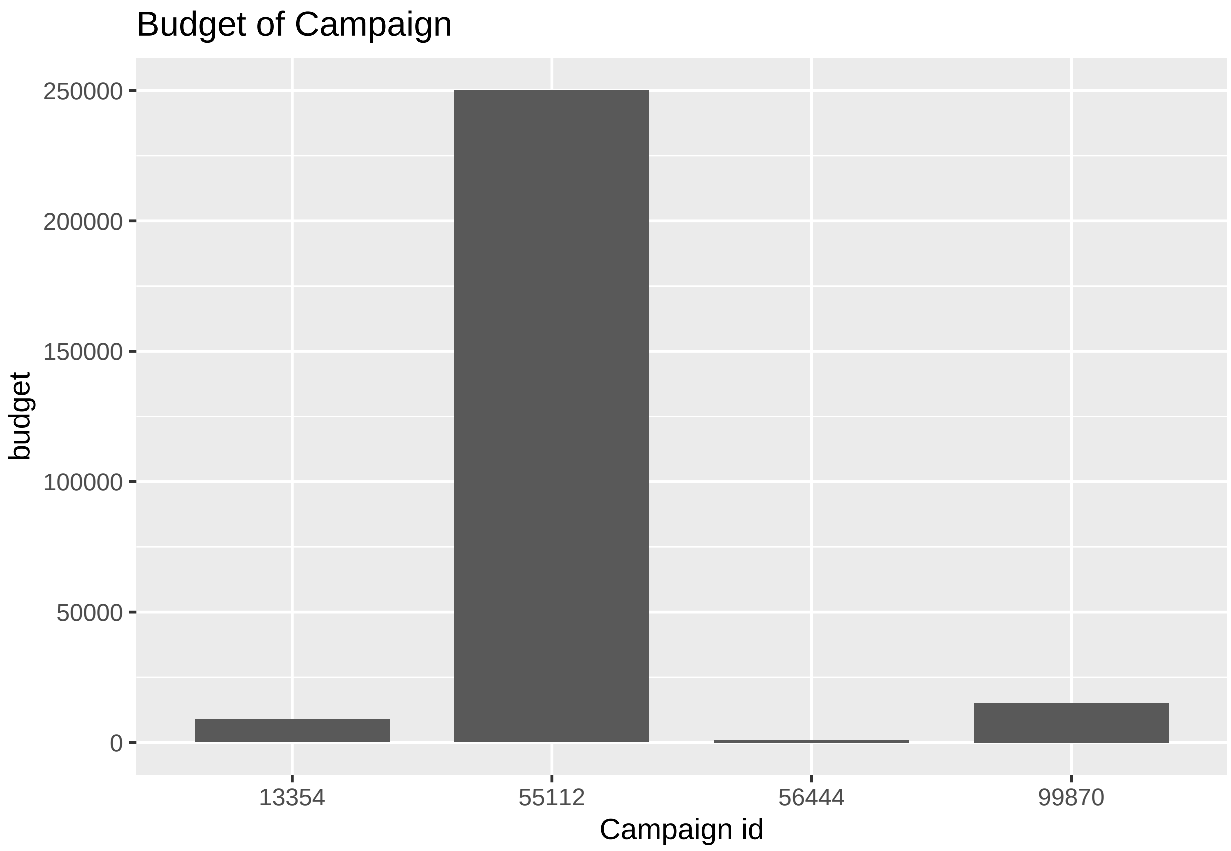
```
clicks_impressions <- select(res,campaign_id,clicks,impressions) %>%
  gather(clicks,impressions,key = type, value = number)

ggplot(data=clicks_impressions) +
  geom_bar(aes(factor(x=campaign_id),y=number,fill=type),stat="identity",position = "dodge")+
  ylab("Count")+xlab("campaign id")
```



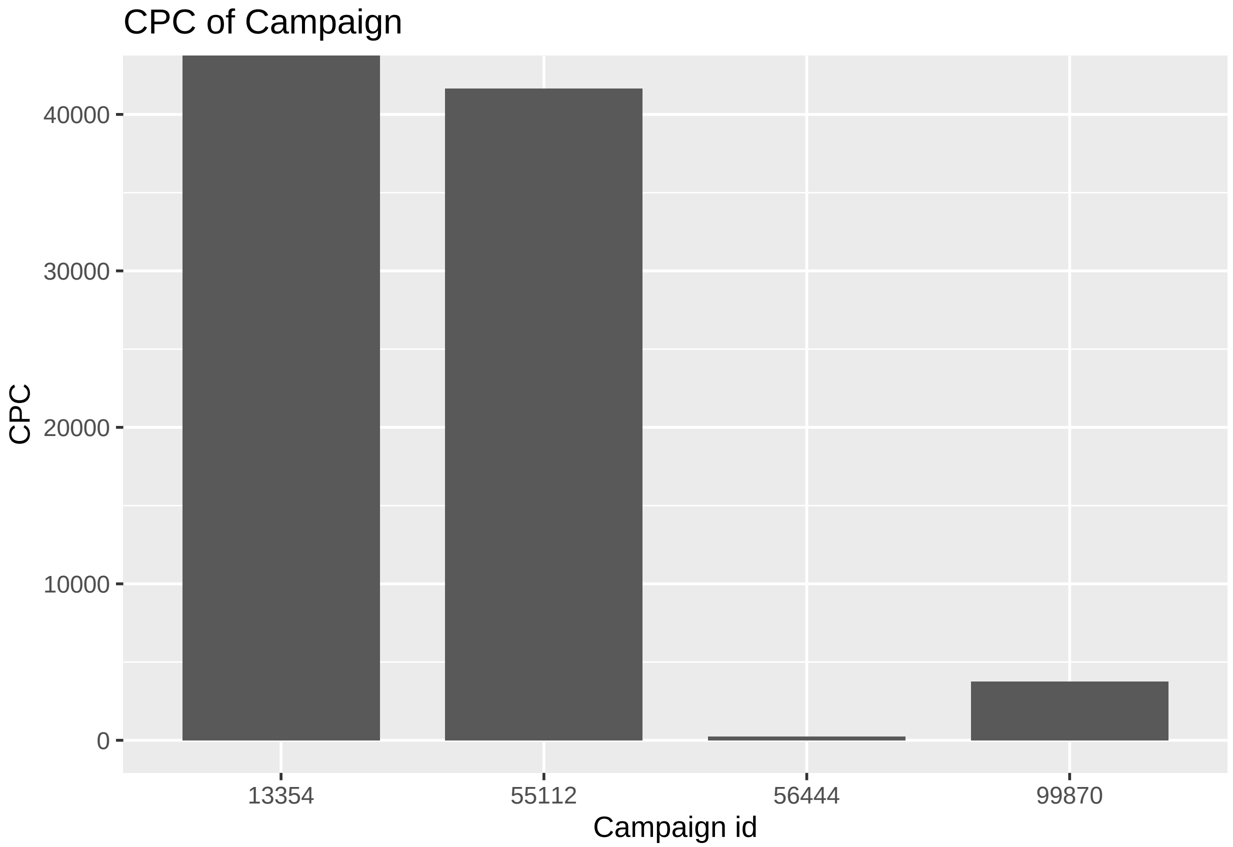
### Budget of each campaign

```
ggplot(data=res, aes(factor(x=campaign_id), y=budget)) +
  geom_bar(stat="identity", width=0.75) +
  ggtitle("Budget of Campaign") +
  xlab("Campaign id")
```



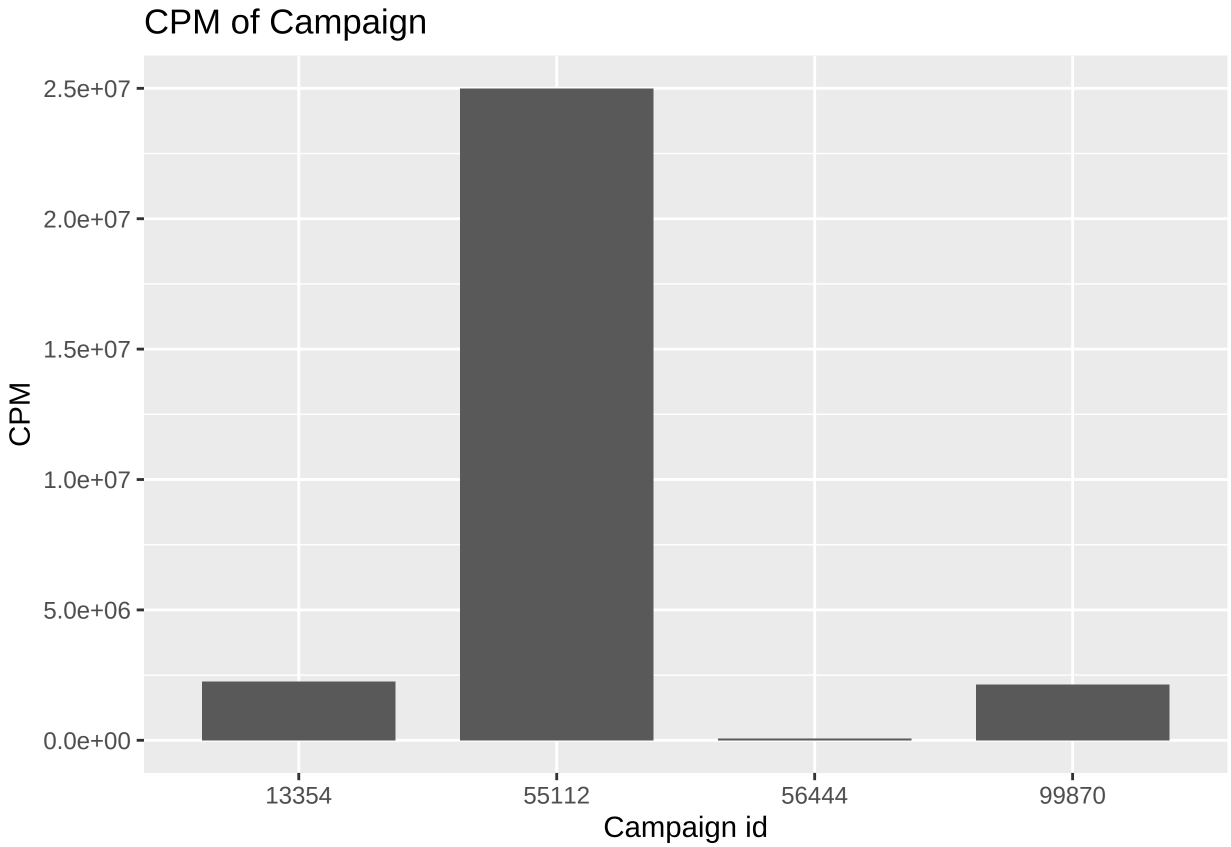
CPC (Cost per click) for each campaign id

```
ggplot(data=res, aes(factor(x=campaign_id), y=CPC)) +
  geom_bar(stat="identity", width=0.75) +
  ggtitle("CPC of Campaign") +
  xlab("Campaign id")
```



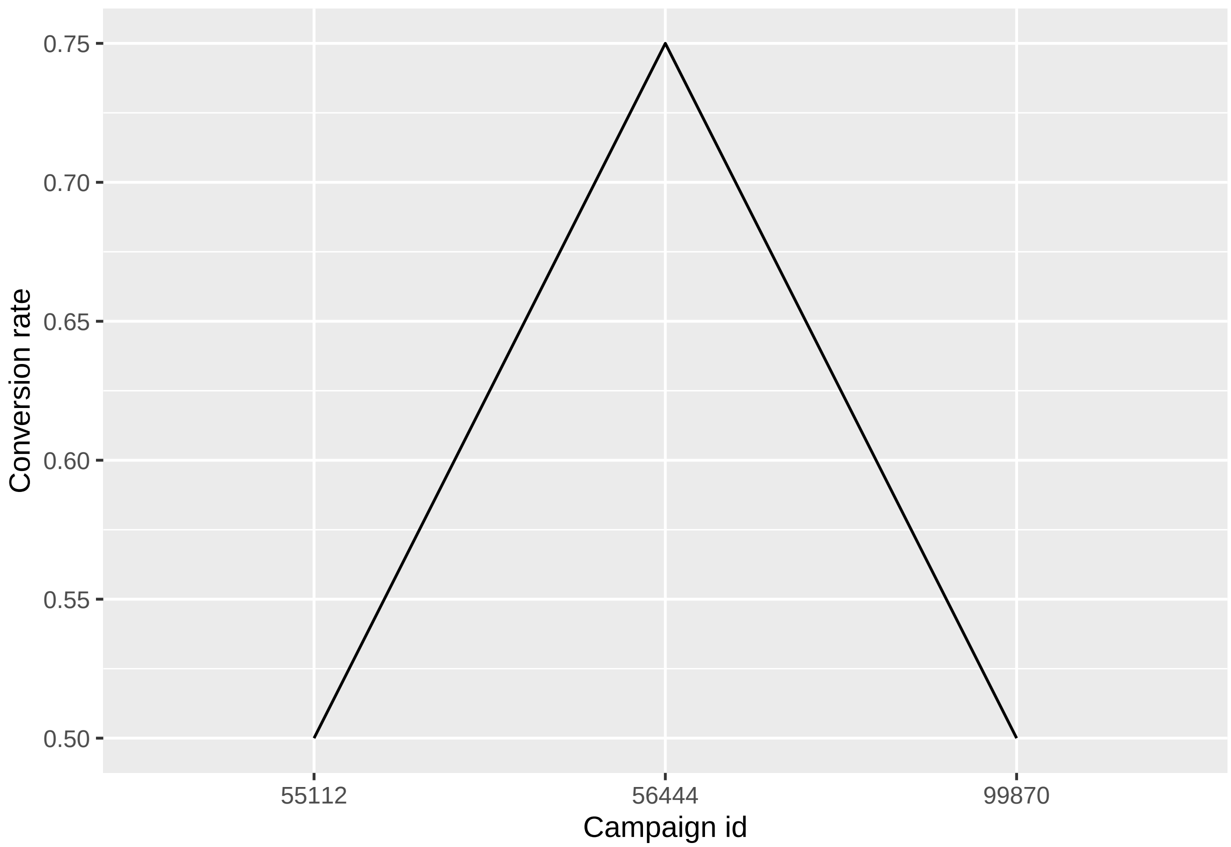
CPM of each campaign

```
ggplot(data=res, aes(factor(x=campaign_id), y=CPM)) +
  geom_bar(stat="identity", width=0.75) +
  ggtitle("CPM of Campaign") +
  xlab("Campaign id")
```



the Convertion rate of each Campaign

```
CR <- select(res,campaign_id,conversion_rate)
CR <- na.omit(CR)
ggplot(data = CR) +
  geom_line(aes(factor(x=campaign_id),y=conversion_rate,group=1)) +
  xlab("Campaign id")+ylab("Conversion rate")
```



```
# From the number of timezone of clicks and conversions, we can see where are the users click the most
count(clicks,timezone)
```

```
##    timezone n
## 1 UTC -5:00 7
## 2 UTC -8:00 4
## 3 UTC +0:00 3
```

```
count(conversions,timezone)
```

```
##    timezone n
## 1 UTC -5:00 3
## 2 UTC -8:00 3
## 3 UTC +0:00 2
```