Club_Mahindra_Code.R

chatr

Fri May 10 11:22:18 2019

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.3.3
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
```

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

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

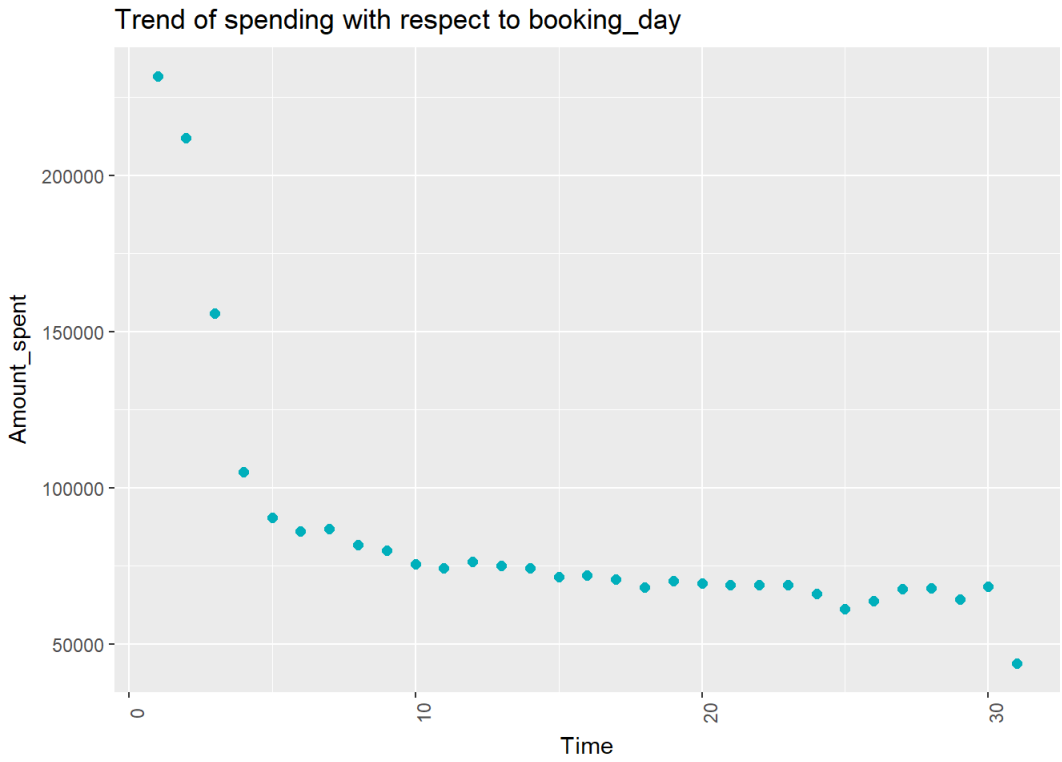
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

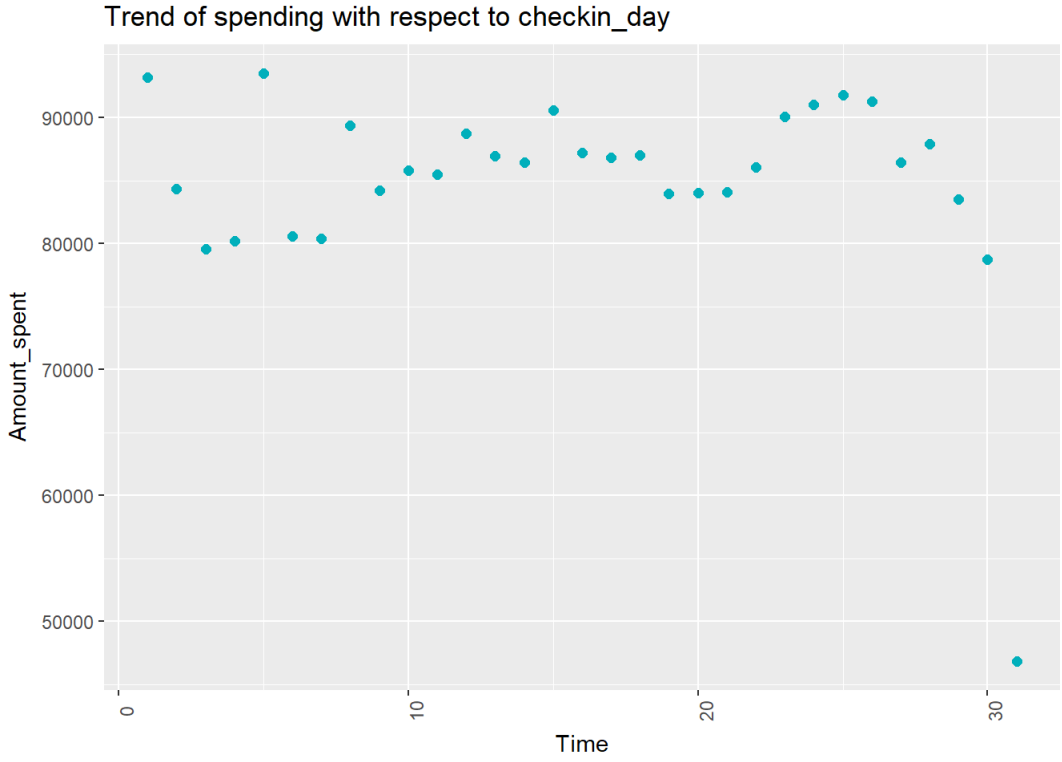
```
train_data <- read.csv("C:/Users/chatr/Desktop/club mahindra/train_5CLrC8b/train.csv")

categorical_cols <- c("reservation_id","channel_code","main_product_code", "persontravellingid",
                      "resort_region_code","resort_type_code","room_type_booked_code","season_holidayed_cod
e",
                      "state_code_residence", "state_code_resort", "member_age_buckets", "booking_type_code
",
                      "memberid", "cluster_code", "reservationstatusid_code", "resort_id" )

train_data[categorical_cols] <- lapply(train_data[categorical_cols], factor)


train_data$booking_date <- dmy(train_data$booking_date)
train_data$checkin_date <- dmy(train_data$checkin_date)
train_data$checkout_date <- dmy(train_data$checkout_date)

################################################
# There are few booking dates from 2012 which is a sign of measurement error
# So, pulling those observations and replacing '2012' with '2018'
# even then booking date falls behind checkin date,
# so replaced booking date with check in date for those observation
################################################
train_data$booking_date[year(train_data$checkin_date) %in% '2012'] <- train_data$checkin_date[year(train_dat
a$checkin_date) %in% '2012']

train_data$checkin_date <- gsub("2012", "2018", train_data$checkin_date)
train_data$booking_date <- gsub("2012", "2018", train_data$booking_date)
train_data$checkout_date <- gsub("2012", "2018", train_data$checkout_date)

train_data$booking_date <- ymd(train_data$booking_date)
train_data$checkin_date <- ymd(train_data$checkin_date)
train_data$checkout_date <- ymd(train_data$checkout_date)

###################################
# calculating the number of days stayed in resort
###################################
train_data$days_stayed <- difftime(train_data$checkout_date, train_data$checkin_date, units = "days")
train_data$time_for_trip <- difftime(train_data$checkin_date, train_data$booking_date, units = "days")

train_data$days_stayed <- as.numeric(train_data$days_stayed)
train_data$time_for_trip <- as.numeric(train_data$time_for_trip)


################################################
# plots to check if any particular day have an impact on amount spent
################################################

temp_data <- mutate(train_data, booking_quarter_days = day(train_data$booking_date),
                checkin_quarter_days = day(train_data$checkin_date),
                checkout_quarter_days = day(train_data$checkout_date))
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

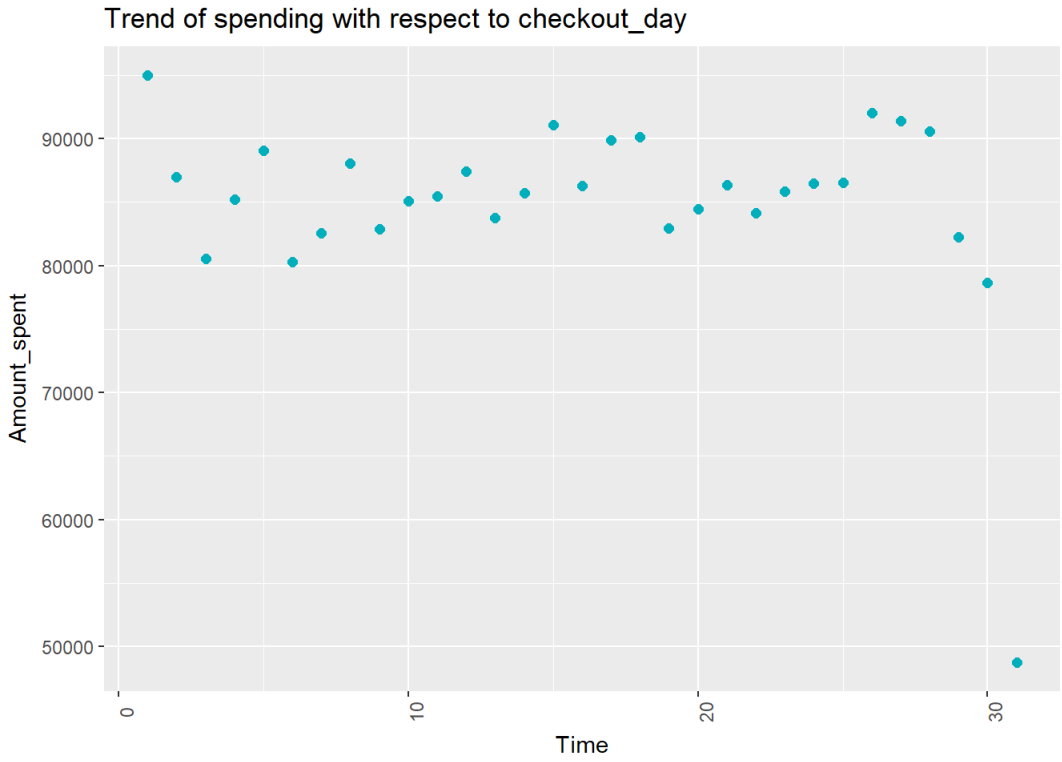
```
# by booking_day
new_data <- aggregate(temp_data["amount_spent_per_room_night_scaled"],
                      by = temp_data["booking_quarter_days"], sum)

ggplot(new_data, mapping = aes(x= new_data$booking_quarter_days,
                               y= new_data$amount_spent_per_room_night_scaled))+
  geom_point(color = "#00AFBB", size = 2)+
  theme(axis.text.x = element_text(angle = 90))+
  ggtitle("Trend of spending with respect to booking_day")+xlab("Time") + ylab("Amount_spent")
```




```
# by checkin_day
new_data <- aggregate(temp_data["amount_spent_per_room_night_scaled"],
                      by = temp_data["checkin_quarter_days"], sum)

ggplot(new_data, mapping = aes(x= new_data$checkin_quarter_days,
                               y= new_data$amount_spent_per_room_night_scaled))+
  geom_point(color = "#00AFBB", size = 2)+
  theme(axis.text.x = element_text(angle = 90))+
  ggtitle("Trend of spending with respect to checkin_day")+xlab("Time") + ylab("Amount_spent")
```



```
# by check_out day
new_data <- aggregate(temp_data["amount_spent_per_room_night_scaled"],
                      by = temp_data["checkout_quarter_days"], sum)

ggplot(new_data, mapping = aes(x= new_data$checkout_quarter_days,
                               y= new_data$amount_spent_per_room_night_scaled))+
  geom_point(color = "#00AFBB", size = 2)+
  theme(axis.text.x = element_text(angle = 90))+
  ggtitle("Trend of spending with respect to checkout_day")+xlab("Time") + ylab("Amount_spent")
```



```
#################################
# continuing the cleaning part
#################################

# split the timestamp to year, month, day for booking_date,checkin_date & checkout_date

train_data <- mutate(train_data,booking_year = year(train_data$booking_date),
                     booking_month = month(train_data$booking_date),
                     booking_actual_day = day(train_data$booking_date),
                     checkin_year = year(train_data$checkin_date),
                     checkin_month = month(train_data$checkin_date),
                     checkin_actual_day = day(train_data$checkin_date),
                     checkout_year = year(train_data$checkout_date),
                     checkout_month = month(train_data$checkout_date),
                     checkout_actual_day = day(train_data$checkout_date))

# extracting day from timestamp (eg: Mon/Tue/Wed,Thr/Fri/Sat/Sun)

train_data <- mutate(train_data,booking_day =weekdays(train_data$booking_date, abbr = TRUE),
                     checkin_day =weekdays(train_data$checkin_date, abbr = TRUE),
                     checkout_day =weekdays(train_data$checkout_date, abbr = TRUE))


########################################
# number of visits
########################################

#####################################################
# I have made a few assumptions that the members made their first @
# @ visit during the time range in which this data is provided.
# Assume for a particular member_id, if their first visit in dataset is Jan 15th,2017.
# Then it is assumed that they didn't use club mahindra services before Jan 15th,2017.
# So, any further visits are considered to be 2nd, 3rd etc...

# So, sorted the timestamp in ascending order
#####################################################

train_data <- train_data[order(train_data$checkin_date, decreasing = F),]
train_data <- train_data %>% group_by(memberid) %>% mutate(number = 1:n())

categorical_cols_2 <- c("booking_year", "booking_month", "booking_actual_day", "checkin_year",
                        "checkin_month", "checkin_actual_day", "checkout_year", "checkout_month",
                        "checkout_actual_day", "booking_day", "checkin_day", "checkout_day" )

train_data[categorical_cols_2] <- lapply(train_data[categorical_cols_2], factor)

#########################
# removing timestamp data and member_id
#########################

temp_cols <- c("booking_date", "checkin_date", "checkout_date","memberid")
cleaned_data <- train_data[, !(colnames(train_data) %in% temp_cols)]

rm(train_data)

# checking for any missing values in data
sapply(cleaned_data, function(x) sum(is.na(x)))
```

```
##                    reservation_id                        channel_code
##                                 0                                   0
##                  main_product_code                      numberofadults
##                                 0                                   0
##                   numberofchildren                   persontravellingid
##                                 0                                   0
##                  resort_region_code                     resort_type_code
##                                 0                                   0
##               room_type_booked_code                          roomnights
##                                 0                                   0
##               season_holidayed_code                 state_code_residence
##                               114                                4764
##                   state_code_resort                           total_pax
##                                 0                                   0
##                  member_age_buckets                   booking_type_code
##                                 0                                   0
##                        cluster_code            reservationstatusid_code
##                                 0                                   0
##                           resort_id amount_spent_per_room_night_scaled
##                                 0                                   0
##                         days_stayed                         time_for_trip
##                                 0                                   0
##                        booking_year                        booking_month
##                                 0                                   0
##                  booking_actual_day                         checkin_year
##                                 0                                   0
##                       checkin_month                  checkin_actual_day
##                                 0                                   0
##                      checkout_year                      checkout_month
##                                 0                                   0
##                 checkout_actual_day                         booking_day
##                                 0                                   0
##                         checkin_day                        checkout_day
##                                 0                                   0
##                              number
##                                 0
```

```
############################################################
# replacing missing values in state_code_residence
############################################################
levels(cleaned_data$state_code_residence) <- c(levels(cleaned_data$state_code_residence), "data_missing")
cleaned_data$state_code_residence <- ifelse(is.na(cleaned_data$state_code_residence),
                                            "data_missing", cleaned_data$state_code_residence)
cleaned_data$state_code_residence <- as.factor(cleaned_data$state_code_residence)


# ###################################
# # ML Model
# ###################################
# library(h2o)
# row.names(cleaned_data) <- 1:nrow(cleaned_data)
#
# index <- sample(1:nrow(cleaned_data), 0.7*nrow(cleaned_data))
# train_data <- cleaned_data[index,]
# test_data <- cleaned_data[-index,]
#
# h2o.init(max_mem_size = "6g")
# train.hex <- as.h2o(train_data)
# test.hex <- as.h2o(test_data)

#############################################
# from here, I have used H2o open source library for
# hyperparameter tuning and used Random Forest, Xgboost & GBM
#############################################


##############################
# End Results
##############################

# GBM turned to out to be best model in my case and I have used grid search to hypertune the parameters
```

```
# Paramters Tuned:
# Parameter                              Value
# nfolds                                     5
# score_tree_interval                        5
# ntrees                                    49
# max_depth                                 15
# min_rows                                  100
# stopping_tolerance              0.0020455244917544505
# distribution                         gaussian
# sample_rate                               0.8
# col_sample_rate                           0.8
# col_sample_rate_per_tree                  0.8
# stopping_metric                      deviance
# (Metric to use for early stopping : logloss for classification, deviance for regression)


# I got a RMSE of 96.550  in public leaderboard and stood at Rank 179
# In private leaderboard RMSE is 97.684 and my rank is 189







##############################
# follow the same steps above for processing AnalyticsVidhya Test data
# which I named as validation data in my case.
##############################

# ##############################
# # validation data
# ##############################
#
# data_1 <- read.csv("C:/Users/chatr/Desktop/club mahindra/test_Jwt0MQH/test.csv")
#
#
# categorical_cols <- c("reservation_id","channel_code","main_product_code", "persontravellingid",
#                       "resort_region_code","resort_type_code","room_type_booked_code","season_holidayed_c
ode",
#                       "state_code_residence", "state_code_resort", "member_age_buckets", "booking_type_co
de",
#                       "memberid", "cluster_code", "reservationstatusid_code", "resort_id" )
#
# data_1[categorical_cols] <- lapply(data_1[categorical_cols], factor)
#
# data_1$booking_date <- dmy(data_1$booking_date)
# data_1$checkin_date <- dmy(data_1$checkin_date)
# data_1$checkout_date <- dmy(data_1$checkout_date)
#
# data_1$booking_date[year(data_1$checkin_date) %in% '2012'] <- data_1$checkin_date[year(data_1$checkin_date
) %in% '2012']
#
# data_1$checkin_date <- gsub("2012", "2018", data_1$checkin_date)
# data_1$booking_date <- gsub("2012", "2018", data_1$booking_date)
# data_1$checkout_date <- gsub("2012", "2018", data_1$checkout_date)
#
#
# data_1$booking_date <- ymd(data_1$booking_date)
# data_1$checkin_date <- ymd(data_1$checkin_date)
# data_1$checkout_date <- ymd(data_1$checkout_date)
#
#
# data_1$days_stayed <- difftime(data_1$checkout_date, data_1$checkin_date, units = "days")
# data_1$time_for_trip <- difftime(data_1$checkin_date, data_1$booking_date, units = "days")
#
# data_1$days_stayed <- as.numeric(data_1$days_stayed)
# data_1$time_for_trip <- as.numeric(data_1$time_for_trip)
#
#
# data_1 <- mutate(data_1,booking_year = year(data_1$booking_date),
#                  booking_month = month(data_1$booking_date),
#                  booking_actual_day = day(data_1$booking_date),
```

```
#                   checkin_year = year(data_1$checkin_date),
#                   checkin_month = month(data_1$checkin_date),
#                   checkin_actual_day = day(data_1$checkin_date),
#                   checkout_year = year(data_1$checkout_date),
#                   checkout_month = month(data_1$checkout_date),
#                   checkout_actual_day = day(data_1$checkout_date))
#
# data_1 <- mutate(data_1,booking_day =weekdays(data_1$booking_date, abbr = TRUE),
#                   checkin_day =weekdays(data_1$checkin_date, abbr = TRUE),
#                       checkout_day =weekdays(data_1$checkout_date, abbr = TRUE) )
#
#
# ########################################
# # number of visits
# ########################################
# data_1 <- data_1[order(data_1$checkin_date, decreasing = F),]
# data_1 <- data_1 %>% group_by(memberid) %>% mutate(number = 1:n())
# # mean_data_by_number <- aggregate(temp_data['amount_spent_per_room_night_scaled'], by = temp_data['number
'], mean )
#
# categorical_cols_2 <- c("booking_year", "booking_month", "booking_actual_day", "checkin_year",
#                         "checkin_month", "checkin_actual_day", "checkout_year", "checkout_month",
#                         "checkout_actual_day", "booking_day", "checkin_day", "checkout_day" )
#
# data_1[categorical_cols_2] <- lapply(data_1[categorical_cols_2], factor)
#
#
# temp_cols <- c("booking_date", "checkin_date", "checkout_date","memberid")
#
# validation_data <- data_1[, !(colnames(data_1) %in% temp_cols)]
#
#
# rm(data_1)
# sapply(validation_data, function(x) sum(is.na(x)))
#
# ########################################################
# # replacing missing values in state_code_residence
# ########################################################
# levels(validation_data$state_code_residence) <- c(levels(validation_data$state_code_residence), "data_miss
ing")
# validation_data$state_code_residence <- ifelse(is.na(validation_data$state_code_residence),
#                                                "data_missing", validation_data$state_code_residence)
# validation_data$state_code_residence <- as.factor(validation_data$state_code_residence)
#
#
# validation.hex <- as.h2o(validation_data)
#
#
# # ###################################
# # # More Plots to check booking, checkin and checkout trend
# # ###################################
# # #
# # #
# # ######################
# # # booking trend
# # ######################
# # # a: by_date
# # booking_data_by_date <- aggregate(train_data["amount_spent_per_room_night_scaled"], by = train_data["boo
king_date"], sum)
# # daily_booking_plot <- ggplot(booking_data_by_date,
# #                              mapping = aes(x= booking_data_by_date$booking_date,
# #                                            y= booking_data_by_date$amount_spent_per_room_night_scaled))
+
# #   geom_point(color = "#00AFBB", size = 2)+
# #   ggtitle("Trend of spending with respect to booking_date by date")+xlab("Time") + ylab("Amount_spent")
# #
# # daily_booking_plot
# # rm(booking_data_by_date)
# #
# # # b: by_month
# # booking_by_month <- train_data[,c("booking_date", "amount_spent_per_room_night_scaled")]
# # booking_by_month <- mutate(booking_by_month,Month_Yr := format(as.Date(booking_by_month$booking_date), "
%Y-%m"))
```

```
# #
# # # setDT(booking_by_month)[, Month_Yr := format(as.Date(booking_by_month$booking_date), "%Y-%m") ]
# #
# # agg_booking_data_month <- aggregate(booking_by_month["amount_spent_per_room_night_scaled"],
# #                                     by = booking_by_month["Month_Yr"], sum)
# #
# # monthly_booking_plot <- ggplot(agg_booking_data_month,
# #                                mapping = aes(x= agg_booking_data_month$Month_Yr,
# #                                              y= agg_booking_data_month$amount_spent_per_room_night_scal
ed))+
# #   geom_point(color = "#00AFBB", size = 2)+theme(axis.text.x = element_text(angle = 90))+
# #   ggtitle("Trend of spending with respect to booking_date by month")+xlab("Time") + ylab("Amount_spent")
# # monthly_booking_plot
# #
# # rm(booking_by_month)
# # rm(agg_booking_data_month)
# #
# # ######################
# # # check_in trend
# # ######################
# # # a: overall
# # checkin_data_by_date <- aggregate(train_data["amount_spent_per_room_night_scaled"],
# #                                   by = train_data["checkin_date"], sum)
# # daily_checkin_plot <- ggplot(checkin_data_by_date,
# #                              mapping = aes(x= checkin_data_by_date$checkin_date,
# #                                            y= checkin_data_by_date$amount_spent_per_room_night_scaled))
+
# #   geom_point(color = "#00AFBB", size = 2)+ggtitle("Trend of spending with respect to checkin_date by dat
e")+xlab("Time") + ylab("Amount_spent")
# # daily_checkin_plot
# #
# # rm(checkin_data_by_date)
# #
# # # b: by_month
# # checkin_by_month <- train_data[,c("checkin_date", "amount_spent_per_room_night_scaled")]
# # checkin_by_month <- mutate(checkin_by_month,Month_Yr := format(as.Date(checkin_by_month$checkin_date), "
%Y-%m"))
# #
# # # setDT(booking_by_month)[, Month_Yr := format(as.Date(booking_by_month$booking_date), "%Y-%m") ]
# #
# # agg_checkin_data_month <- aggregate(checkin_by_month["amount_spent_per_room_night_scaled"],
# #                                     by = checkin_by_month["Month_Yr"], sum)
# #
# # monthly_checkin_plot <- ggplot(agg_checkin_data_month,
# #                                mapping = aes(x= agg_checkin_data_month$Month_Yr,
# #                                              y= agg_checkin_data_month$amount_spent_per_room_night_scal
ed))+
# #   geom_point(color = "#00AFBB", size = 2)+theme(axis.text.x = element_text(angle = 90))+ggtitle("Trend o
f spending with respect to checkin_date by Month")+xlab("Time") + ylab("Amount_spent")
# #
# # monthly_checkin_plot
# # rm(agg_checkin_data_month)
# # rm(checkin_by_month)
# #
# # ######################
# # # check_out trend
# # ######################
# # # a: overall
# # checkout_data_by_date <- aggregate(train_data["amount_spent_per_room_night_scaled"],
# #                                    by = train_data["checkout_date"], sum)
# # daily_checkout_plot <- ggplot(checkout_data_by_date,
# #                               mapping = aes(x= checkout_data_by_date$checkout_date,
# #                                             y= checkout_data_by_date$amount_spent_per_room_night_scaled
))+
# #   geom_point(color = "#00AFBB", size = 2)+ggtitle("Trend of spending with respect to checkout_date by da
te")+xlab("Time") + ylab("Amount_spent")
# # daily_checkout_plot
# #
# # rm(checkout_data_by_date)
# #
# # # b: by_month
# # checkout_by_month <- train_data[,c("checkout_date", "amount_spent_per_room_night_scaled")]
# # checkout_by_month <- mutate(checkout_by_month,Month_Yr := format(as.Date(checkout_by_month$checkout_date
```

```
# # checkout_by_month <- mutate(checkout_by_month,Month_Yr = format(as.Date(checkout_by_month$checkout_date
), "%Y-%m"))
# #
# # # setDT(booking_by_month)[, Month_Yr := format(as.Date(booking_by_month$booking_date), "%Y-%m") ]
# #
# # agg_checkout_data_month <- aggregate(checkout_by_month["amount_spent_per_room_night_scaled"],
# #                                      by = checkout_by_month["Month_Yr"], sum)
# #
# # monthly_checkout_plot <- ggplot(agg_checkout_data_month,
# #                                 mapping = aes(x= agg_checkout_data_month$Month_Yr,
# #                                               y= agg_checkout_data_month$amount_spent_per_room_night_sc
aled))+
# #   geom_point(color = "#00AFBB", size = 2)+theme(axis.text.x = element_text(angle = 90))+ggtitle("Trend o
f spending with respect to checkout_date by Month")+xlab("Time") + ylab("Amount_spent")
# #
# # monthly_checkout_plot
# #
# # rm(agg_checkout_data_month)
# # rm(checkout_by_month)
```