Descriptive and Predictive Data: Final Exam Report

Quang Phong - 6286943

2022-06-10

DATA LOADING AND UNDERSTANDING

```
# Load data
load("gamedata.Rdata")
```

```
# Rename the data frame as "df_Game" for name convention consistency
df_Game <- gamedata
rm(gamedata)
```

```
# Have a quick look at the data
head(df_Game)
```

```
##   spending monthindex playerindex    genre  motivation       type gender
## 1 155.8784          1           1     misc destruction individual female
## 2 161.9487          2           1     misc destruction individual female
## 3 109.9797          3           1 strategy destruction individual female
## 4 134.9355          4           1   puzzle destruction individual female
## 5 124.8536          5           1   puzzle destruction individual female
## 6 146.0078          6           1     misc destruction individual female
##     income age dummydecember
## 1 1952.377  19             0
## 2 1772.635  19             0
## 3 1651.704  19             0
## 4 1911.066  19             0
## 5 1539.180  19             0
## 6 1667.979  19             0
```

QUESTION 1

```
# Print summary statistics of all variables in the dataset
summary(df_Game)
```

```
##     spending       monthindex      playerindex           genre
##  Min.   :  0.0   Min.   :  1.00   Min.   :  1.0   misc      :12686
##  1st Qu.:233.7   1st Qu.: 30.75   1st Qu.:125.8   platform  : 8272
##  Median :373.1   Median : 60.50   Median :250.5   puzzle    :10676
```

```
##  Mean   :371.3   Mean   : 60.50   Mean   :250.5   shooter   : 7205
##  3rd Qu.:503.6   3rd Qu.: 90.25   3rd Qu.:375.2   simulation:11729
##  Max.   :749.1   Max.   :120.00   Max.   :500.0   strategy  : 9432
##        motivation            type          gender          income
##  competition:35400   individual:22680   female:24000   Min.   : 762
##  destruction:24600   social    :37320   male  :36000   1st Qu.:2098
##                                                        Median :2722
##                                                        Mean   :2602
##                                                        3rd Qu.:3066
##                                                        Max.   :4118
##       age        dummydecember
##  Min.   :18.00   Min.   :0.00000
##  1st Qu.:29.00   1st Qu.:0.00000
##  Median :36.00   Median :0.00000
##  Mean   :36.15   Mean   :0.08333
##  3rd Qu.:43.00   3rd Qu.:0.00000
##  Max.   :55.00   Max.   :1.00000
```

```
str(df_Game)
```

```
## 'data.frame':    60000 obs. of  10 variables:
##  $ spending     : num  156 162 110 135 125 ...
##  $ monthindex   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ playerindex  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ genre        : Factor w/ 6 levels "misc","platform",..: 1 1 6 3 3 1 1 1 3 1 ...
##  $ motivation   : Factor w/ 2 levels "competition",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ type         : Factor w/ 2 levels "individual","social": 1 1 1 1 1 1 1 1 1 1 ...
##  $ gender       : Factor w/ 2 levels "female","male": 1 1 1 1 1 1 1 1 1 1 ...
##  $ income       : num  1952 1773 1652 1911 1539 ...
##  $ age          : num  19 19 19 19 19 19 19 19 20 20 ...
##  $ dummydecember: num  0 0 0 0 0 0 0 0 0 0 ...
```

Comment: - There are 10 variables in this dataset of 60000 observations. - Categorical variables include: genre, motivation, type, gender, dummydecember. They have finite values. - Continuous variables include: spending, income. - Discrete variables are: monthindex, playerindex, age. Technically, if age takes any value (e.g., 15.78 years old), it is considered continuous. However, here age only takes integer values, so it is considered discrete variable. - The youngest person in the research period was 18 years old at the beginning, and the oldest turned 55 at the end. - 60% of the players are male and 40% are female, which is quite a balanced ratio.

```
# Convert categorical to factors
df_Game$dummydecember <- as.factor(df_Game$dummydecember)
df_Game$genre <- as.factor(df_Game$genre)
df_Game$motivation <- as.factor(df_Game$motivation)
```

```
# Check the types of variables again
str(df_Game)
```

```
## 'data.frame':    60000 obs. of  10 variables:
##  $ spending     : num  156 162 110 135 125 ...
##  $ monthindex   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ playerindex  : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ genre       : Factor w/ 6 levels "misc","platform",..: 1 1 6 3 3 1 1 1 3 1 ...
##  $ motivation  : Factor w/ 2 levels "competition",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ type        : Factor w/ 2 levels "individual","social": 1 1 1 1 1 1 1 1 1 1 ...
##  $ gender      : Factor w/ 2 levels "female","male": 1 1 1 1 1 1 1 1 1 1 ...
##  $ income      : num  1952 1773 1652 1911 1539 ...
##  $ age         : num  19 19 19 19 19 19 19 19 20 20 ...
##  $ dummydecember: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

Comment: - Now each variable has been assigned to its correct data type.

```r
# Check the number of gamers
length(unique(df_Game$playerindex))
```

```
## [1] 500
```

Comment: There are 500 games in the dataset.

```r
# Check the number of months
length(unique(df_Game$monthindex))
```

```
## [1] 120
```

Comment: There are 120 months, corresponding to 10 years in the dataset.

QUESTION 2

2A

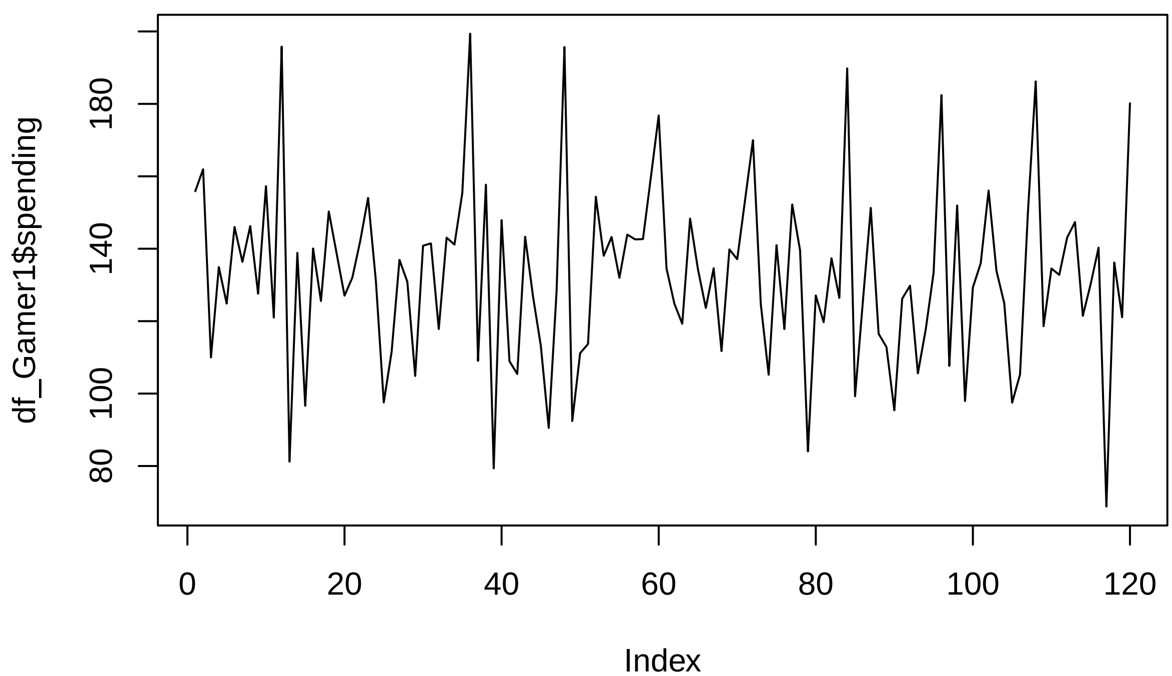
```r
# Choose all time series data for gamer ID 1 [INCOME, SPENDING]
df_Gamer1 <- df_Game[which(df_Game$playerindex == 1), ]
```

```r
# Print summary of the selected dataset
summary(df_Gamer1)
```

```
##     spending       monthindex      playerindex        genre
##  Min.   : 68.8   Min.   :  1.00   Min.   :1    misc      :32
##  1st Qu.:117.8   1st Qu.: 30.75   1st Qu.:1    platform  : 9
##  Median :133.6   Median : 60.50   Median :1    puzzle    :23
##  Mean   :132.5   Mean   : 60.50   Mean   :1    shooter   : 7
##  3rd Qu.:143.5   3rd Qu.: 90.25   3rd Qu.:1    simulation:32
##  Max.   :199.4   Max.   :120.00   Max.   :1    strategy  :17
##        motivation           type         gender        income          age
##  competition:  0   individual:120   female:120   Min.   :1096   Min.   :19.00
##  destruction:120   social    :  0   male  :  0   1st Qu.:1773   1st Qu.:21.00
##                                                  Median :1988   Median :24.00
##                                                  Mean   :1963   Mean   :23.83
##                                                  3rd Qu.:2146   3rd Qu.:26.00
##                                                  Max.   :2677   Max.   :29.00
```

```
##   dummydecember
##   0:110
##   1: 10
##
##
##
##
```

```r
# Plot the spending of this gamer over 10 years
plot(df_Gamer1$spending, type = "l")
```



Comment on the characteristics of this gamer: - This player is a female, belonging to 40% female players in 500 players here. - She was 19 years old at the start of the research period and became 29 at the end. This means she was younger than an average player in this data set. - Motivation to play of this player was always for destruction, which is different from the majority of observations in the dataset who has competition motivation. - She never played any game of social type, which is unlike the majority of observations in the dataset who likes to play social games. - In terms of mean income, she earned 1963, 24.6% less than the average of all players (2602). - Regarding mean spending, she spent 132.5, 64.3% less than the average of all players (371.3). - Looking at the plot of spending, we can see the amount of money she spent on gaming fluctuated and did not stay the same for every month. It seemed to have a seasonality of 12 months.

2B

```r
# Step 1: Check the stationarity of the spending data for gamer 1
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
adf.test(df_Gamer1$spending)
```

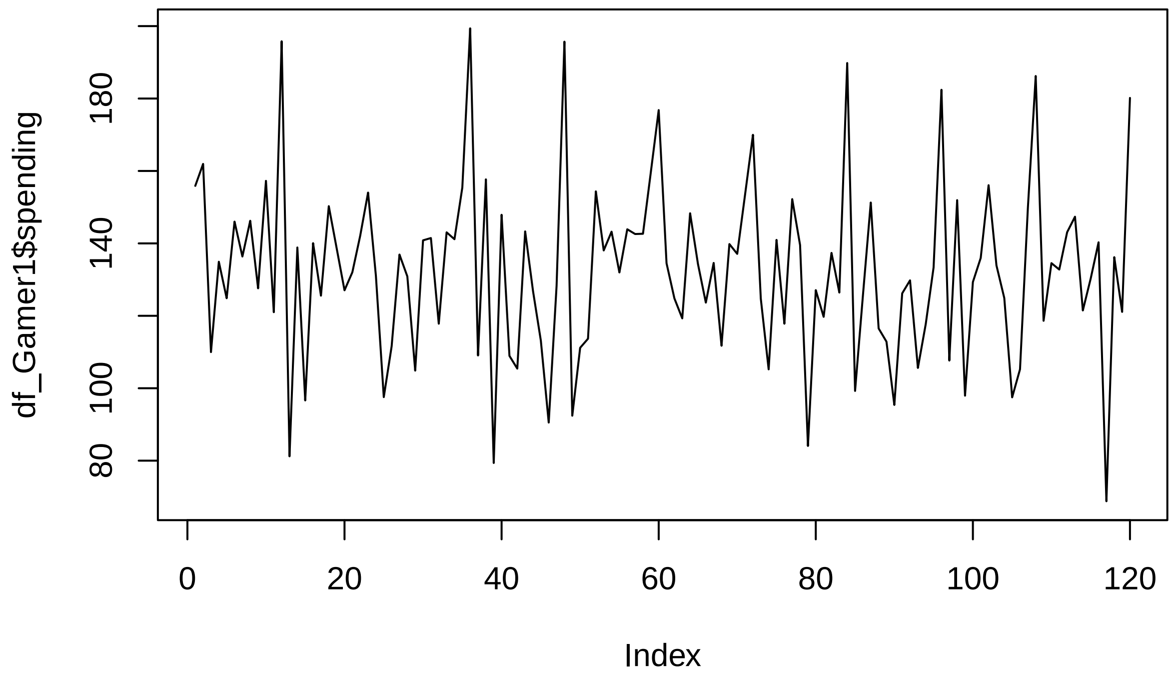
```
##
##  Augmented Dickey-Fuller Test
##
## data:  df_Gamer1$spending
## Dickey-Fuller = -4.9002, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(df_Gamer1$spending)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  df_Gamer1$spending
## KPSS Level = 0.088869, Truncation lag parameter = 4, p-value = 0.1
```

Comment: - We reject Null Hypothesis (H0: the time series is non-stationary) in Unit Root Test (ADF Test) because the p-value = 0.01 < 0.05. - Also, we cannot reject Null Hypothesis (H0: the time series is stationary) in KPSS Test as the p-value = 0.1 > 0.05. - Therefore, both of the tests support that the spending over time for gamer 1 is stationary over time.

```r
# Step 2: Check the seasonality

# Regarding time series properties of gamer 1's spending, initially, we will again look at the plot of 
plot(df_Gamer1$spending, type = "l")
```



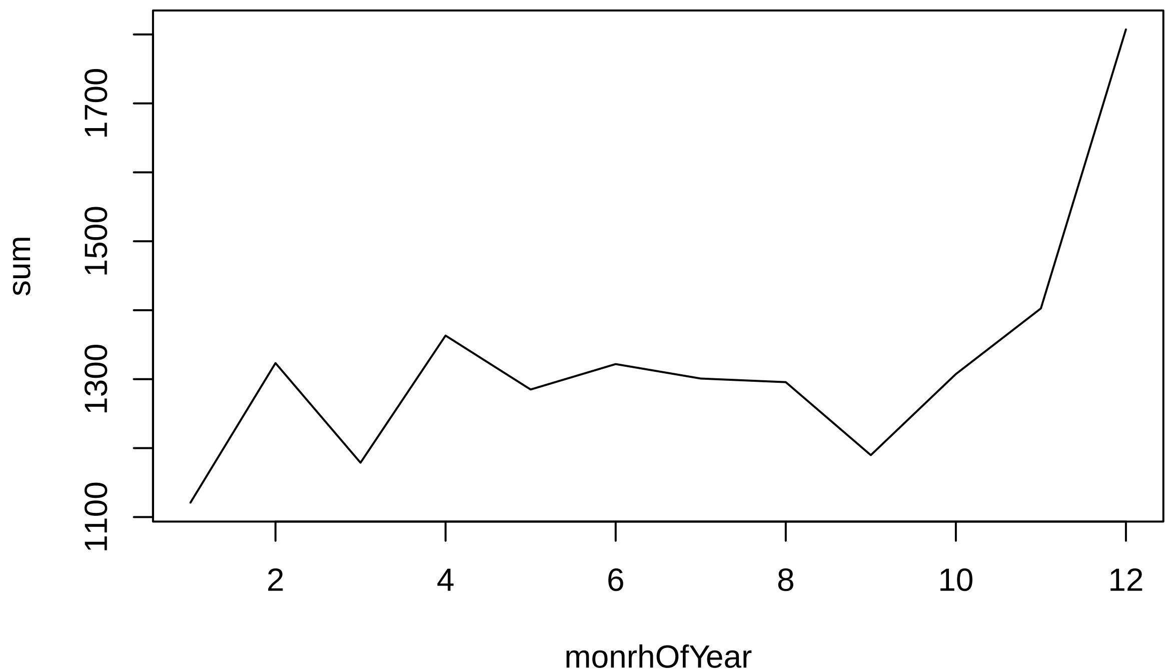
Comment: - By simply looking at this plot, I estimate that there can be seasonality of 12 months, but possibly no trend.

```
# To corroborate this estimation, I will plot the monthly means of spending of this player over 10 years

# Let's create month of year variable so that we can group by month of year later
library("dplyr")
```

```
## Warning: package 'dplyr' was built under R version 4.1.2

##
## 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
```

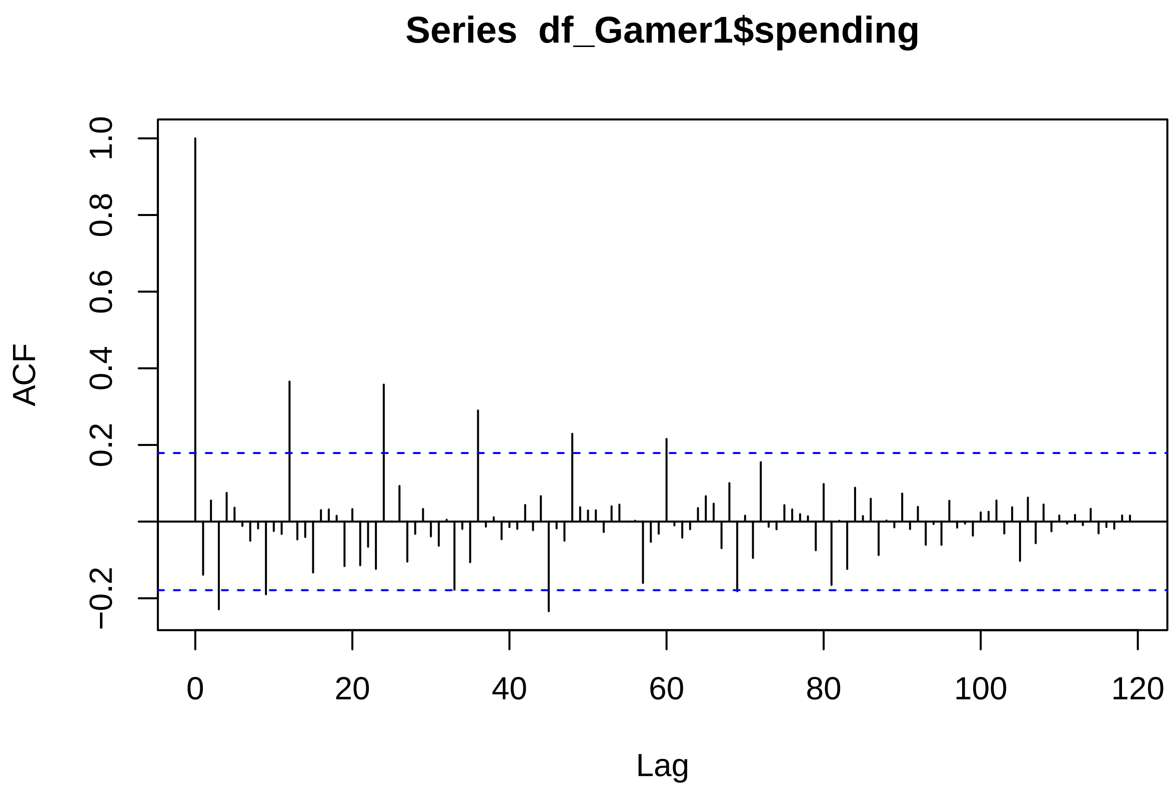
```
df_Gamer1$monrhOfYear <- ifelse(as.numeric(df_Gamer1$monthindex) %% 12 != 0,
                                as.numeric(df_Gamer1$monthindex) %% 12,
                                12)
plot(df_Gamer1 %>%
```

```
      group_by(monrhOfYear) %>%
      summarise(sum = sum(spending)),
    type = "l")
```

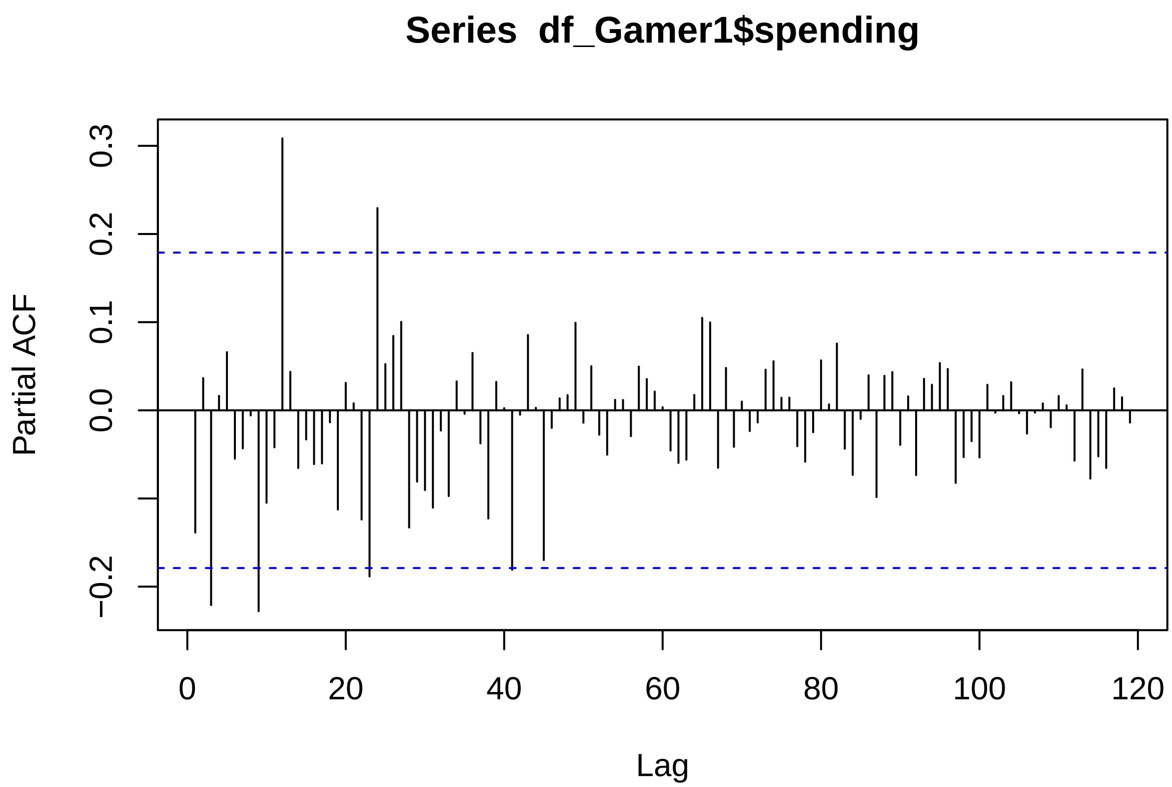


Comment: - As seen from the graph, the average spending in month 12 of 10 years year is higher than the rest.

```
# However, additional statistical evidence must be exhibited.

# Then I will look at ACF and PACF plots.
acf(df_Gamer1$spending, lag.max = 120)
```

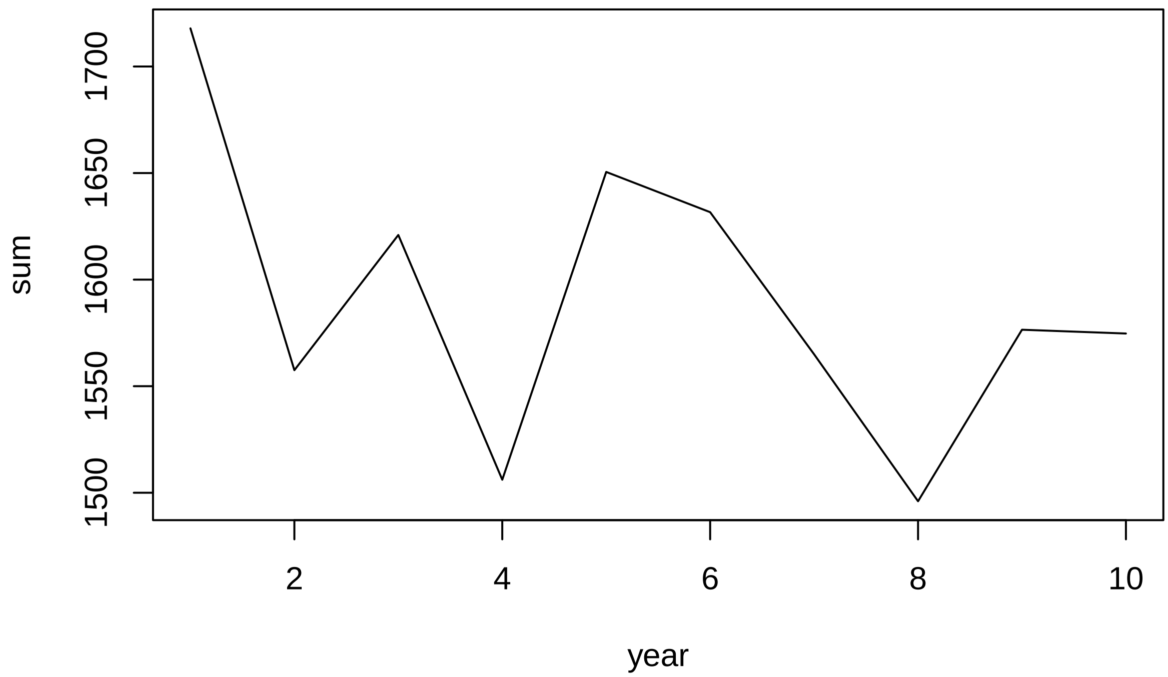



```
pacf(df_Gamer1$spending, lag.max = 120)
```



Comment: - It is easy to see the existence of autocorrelation between lagged spendings here, illustrated by the lines passing the threshold. - From the ACF and PACF plots, it can be concluded there is a seasonality of 12 months in the spending of this gamer. - In details, the autocorrelation at lag 12 is significantly high and gradually geometrically decays at lag 24, 36, 48, 60, etc. Also, we have high partial autocorrelation at lag 12. - Therefore, it is likely that we have seasonal AR (ARIMA (0,0,0),(1,0,0)12) model for this spending of player 1.

```
# Step 3: Check the trend

# First, I will plot the annual spending of gamer 1.
# Let's create year variable so that we can group by year later
df_Gamer1$year <- (as.numeric(df_Gamer1$monthindex)-1) %/% 12 + 1

plot(df_Gamer1 %>%
       group_by(year) %>%
       summarise(sum = sum(spending)),
     type = "l")
```



Comment: - The spending decreased after 10 years, but there is no consistent downwards trend over time.

```
# Second, we can further check the trend by regression
model1 <- lm(df_Gamer1$spending ~ df_Gamer1$monthindex)
summary(model1)
```

```
##
## Call:
## lm(formula = df_Gamer1$spending ~ df_Gamer1$monthindex)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -61.49 -13.58   1.92  11.29  65.96
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           134.81928    4.48627  30.052   <2e-16 ***
## df_Gamer1$monthindex  -0.03877    0.06435  -0.603    0.548
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.42 on 118 degrees of freedom
## Multiple R-squared:  0.003067,   Adjusted R-squared:  -0.005382
## F-statistic: 0.363 on 1 and 118 DF,  p-value: 0.548
```

Comment: - As the impact of monthindex on spending is not statistically significant, there is no trend in the data for gamer 1.

2C

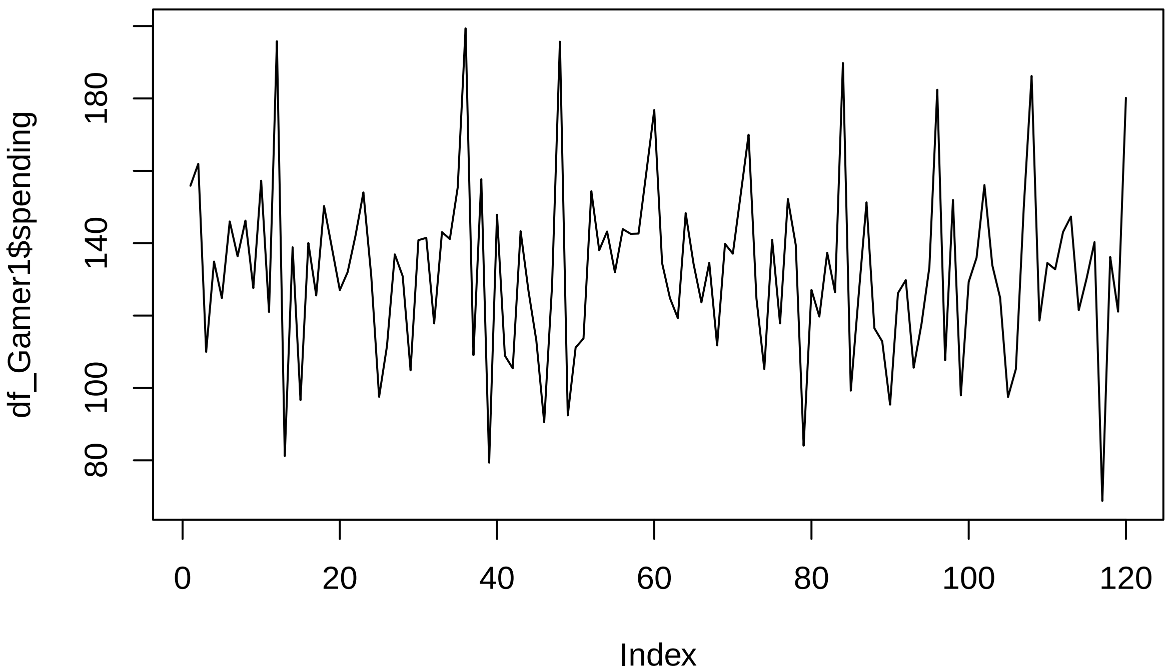
```r
# Print summary of the selected dataset
summary(df_Gamer1)
```

```
##     spending       monthindex      playerindex       genre
##  Min.   : 68.8   Min.   :  1.00   Min.   :1    misc      :32
##  1st Qu.:117.8   1st Qu.: 30.75   1st Qu.:1    platform  : 9
##  Median :133.6   Median : 60.50   Median :1    puzzle    :23
##  Mean   :132.5   Mean   : 60.50   Mean   :1    shooter   : 7
##  3rd Qu.:143.5   3rd Qu.: 90.25   3rd Qu.:1    simulation:32
##  Max.   :199.4   Max.   :120.00   Max.   :1    strategy  :17
##        motivation          type         gender        income           age
##  competition:  0   individual:120   female:120   Min.   :1096   Min.   :19.00
##  destruction:120   social    :  0   male  :  0   1st Qu.:1773   1st Qu.:21.00
##                                                  Median :1988   Median :24.00
##                                                  Mean   :1963   Mean   :23.83
##                                                  3rd Qu.:2146   3rd Qu.:26.00
##                                                  Max.   :2677   Max.   :29.00
##  dummydecember  monrhOfYear         year
##  0:110         Min.   : 1.00   Min.   : 1.0
##  1: 10         1st Qu.: 3.75   1st Qu.: 3.0
##                Median : 6.50   Median : 5.5
##                Mean   : 6.50   Mean   : 5.5
##                3rd Qu.: 9.25   3rd Qu.: 8.0
##                Max.   :12.00   Max.   :10.0
```

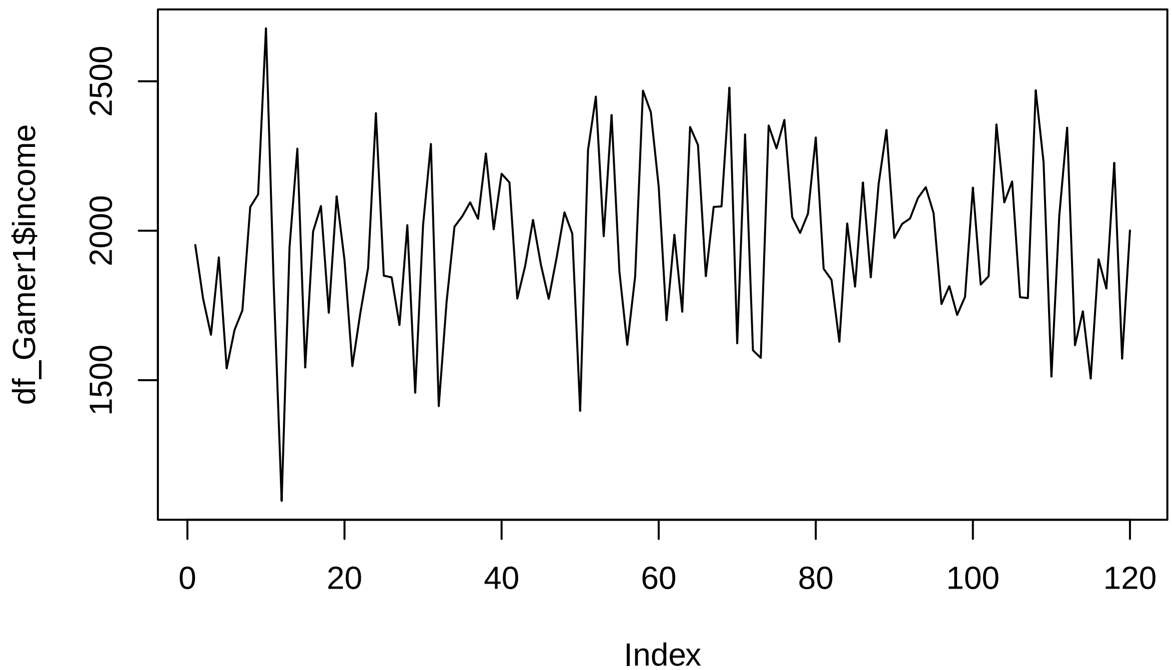
Comment: - This player is a female, belonging to 40% female players in 500 players here. - She was 19 years old at the start of the research period and became 29 at the end. This means she was younger than an average player in this data set. - Motivation to play of this player was always for destruction, which is different from the majority of observations in the dataset who has competition motivation. - She never played any game of social type, which is unlike the majority of observations in the dataset who likes to play social games. - In terms of mean income, she earned 1963, 24.6% less than the average of all players (2602). - Regarding mean spending, she spent 132.5, 64.3% less than the average of all players (371.3). - Looking at the plot of spending, we can see the amount of money she spent on gaming fluctuated and did not stay the same for every month. It seemed to have a seasonality of 12 months.

Variables that change over time: spending, income, genre, age

```r
# Plot each variable over time for gamer 1 (only changing variables)
plot(df_Gamer1$spending, type = "l")
```

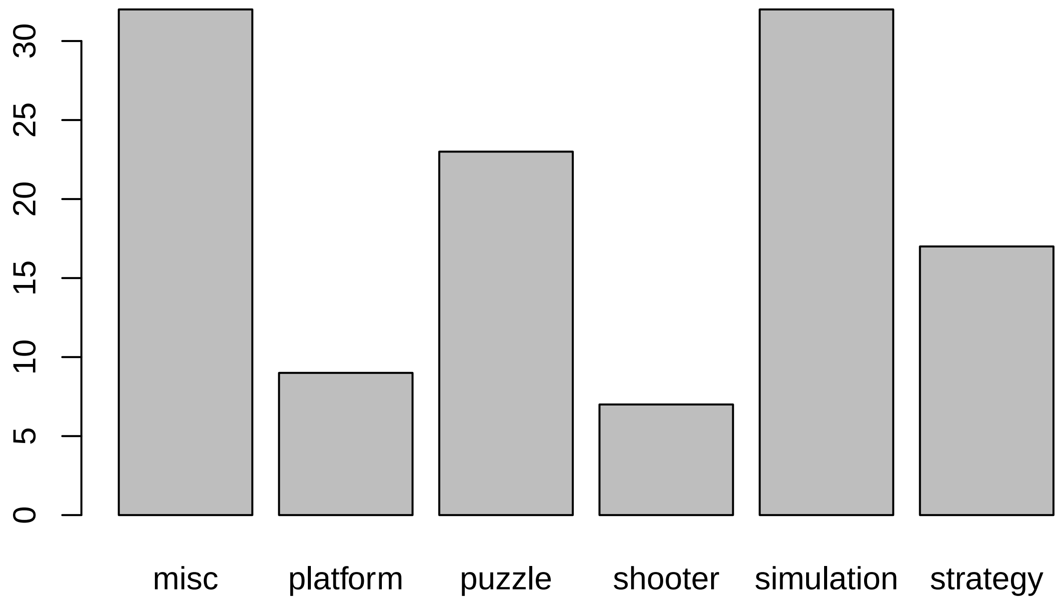



```
plot(df_Gamer1$income, type = "l")
```

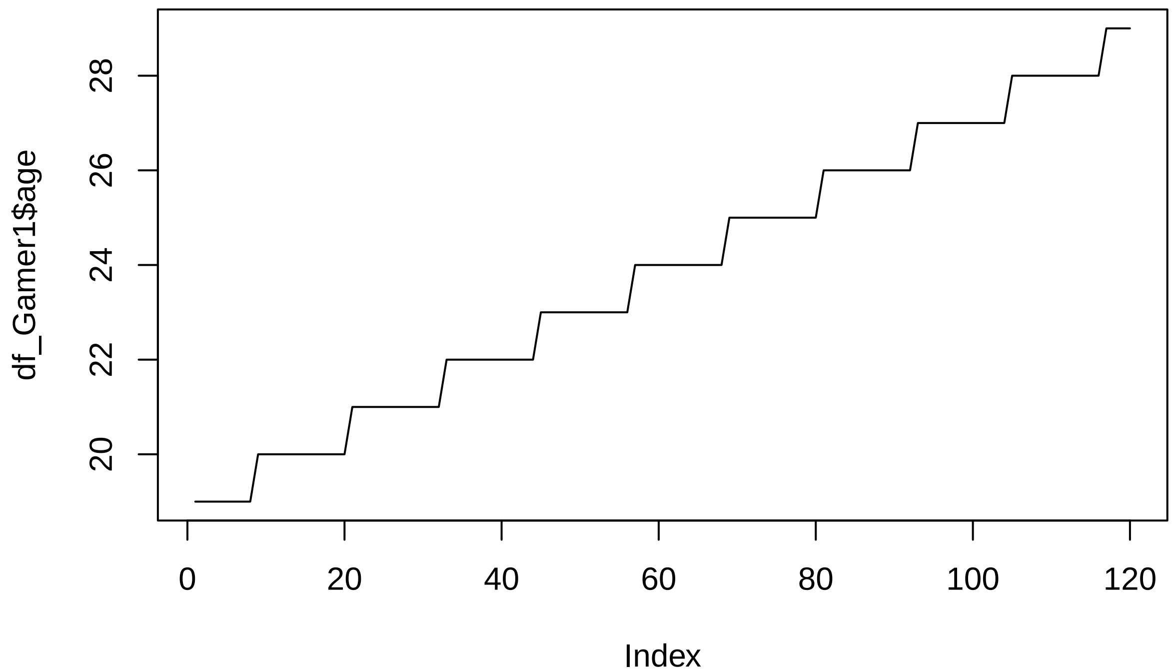



```
plot(df_Gamer1$genre, type = "l")
```

```
## Warning in plot.window(xlim, ylim, log = log, ...): graphical parameter "type"
## is obsolete

## Warning in axis(if (horiz) 2 else 1, at = at.l, labels = names.arg, lty =
## axis.lty, : graphical parameter "type" is obsolete

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## graphical parameter "type" is obsolete

## Warning in axis(if (horiz) 1 else 2, cex.axis = cex.axis, ...): graphical
## parameter "type" is obsolete
```




```
plot(df_Gamer1$age, type = "l")
```



2D

```r
# Estimate a regression model
model2 <- lm(spending ~ income + genre +  age + dummydecember - 1, data = df_Gamer1)
summary(model2)
```

```
##
## Call:
## lm(formula = spending ~ income + genre + age + dummydecember -
##     1, data = df_Gamer1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.800  -4.724  -0.821   5.826  18.675
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## income             0.012012   0.003113   3.859 0.000192 ***
## genremisc        136.457504   8.854042  15.412  < 2e-16 ***
## genreplatform     97.871283   9.737207  10.051  < 2e-16 ***
## genrepuzzle      119.938266   9.111049  13.164  < 2e-16 ***
## genreshooter      73.797897   9.453856   7.806 3.54e-12 ***
## genresimulation 127.647593   9.155328  13.942  < 2e-16 ***
## genrestrategy    103.866620   9.137124  11.368  < 2e-16 ***
```

```
## age               -0.636950   0.302848  -2.103 0.037708 *
## dummydecember1   51.643412   3.133089  16.483  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.285 on 111 degrees of freedom
## Multiple R-squared:  0.9956, Adjusted R-squared:  0.9952
## F-statistic:  2793 on 9 and 111 DF,  p-value: < 2.2e-16
```

Comment: - Based on the results, there are 4 variables that statistically significantly explain the spending of gamer 1. - dummydecember has a positive effect on the spending with p-value < 0.05. In December, the player invested on average 51.64 higher than each of the other months, keeping other variables constant. - genre also has significant effects on the spending with p-value < 0.05. The player spends most when playing misc genre, followed by simulation, puzzle, strategy, platform, and finally shooter. - Income also has a positive effect on the spending although the magnitude is small (0.01) - Lastly, age has a negative effect at 0.05 significant level. - The Adjusted R-squared of the model is 0.8546, which means the model is able to explains 85.5% the variance of player 1's spending over 120 months.

2E

There are several disadvantages of an "average regression" instead of multi-level regression analysis.

- First, because we take average of every explanatory variables and also the explained variable, there will be information loss. For example, the increases and decreases in one variable after a period of all players can balance each others, resulting in an average variable that did not change much over time. Also here, we are interested in the effects of lagged spending variables, and we also see the seasonality in spending of gamer 1. But if we do use average spending of all players, it can be harder to detect them as there will be more noises, except for the case where all players share the same seasonality patterns.
- Second, without multi-level regression analysis, it is challenging to take into account the behaviors of some players. For example, here gamer 1 spent 63.4% less than the average. If it is smaller, the average regression will be leaned to the bigger variances in spending of other players with higher spending.
- Third, a disadvantage of this "average regression" is that you cannot use time-invariant variables to explain the spending. For example, the gender of the "average player" will be fixed over time, so it will not be included in an "average model". The average model will be for a fixed gender of "0.4 female, 0.6 male" and does not take into account possible effects of different genders on spendings. However, if we makes several typical profiles and run regression for each, we can compare and see the effects of these unique characteristics of a group on spending.
- Therefore, it is better to divide our players to several groups (typical profiles) based on genders, motivation, etc. then run different regression models for these groups.

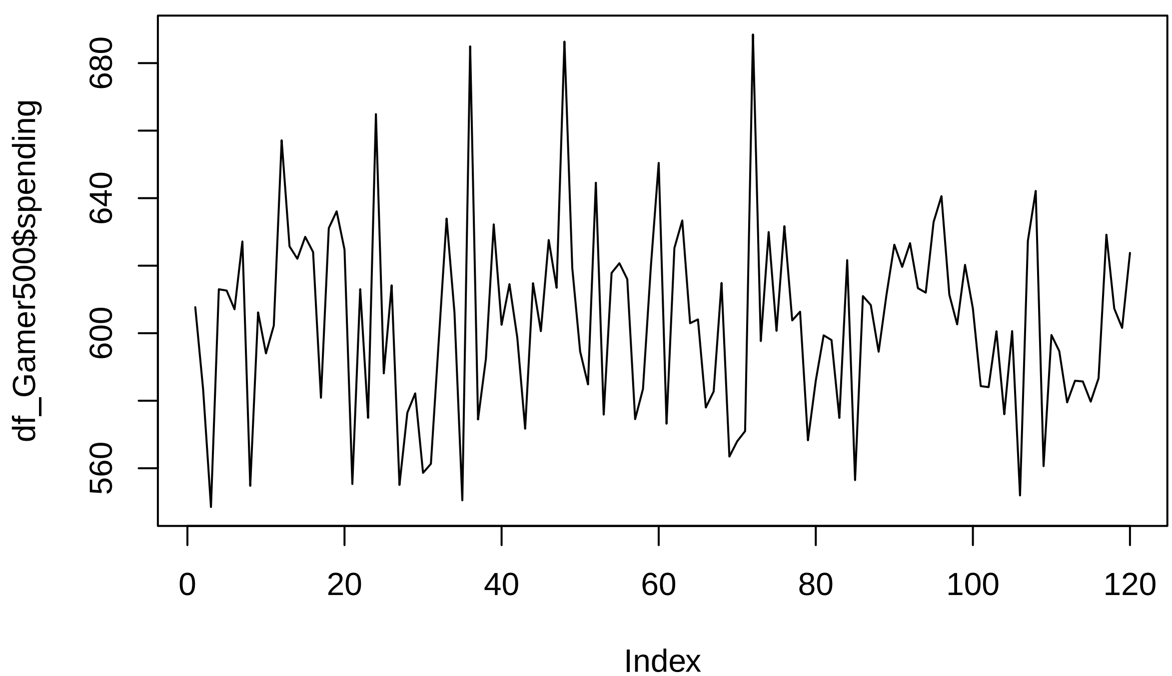
QUESTION 3

3A

```r
# Choose time series data for gamer ID 500
df_Gamer500 <- df_Game[which(df_Game$playerindex == 500), ]

# Print summary of the selected dataset
summary(df_Gamer500)
```

```
##     spending       monthindex      playerindex         genre
##  Min.   :548.5   Min.   :  1.00   Min.   :500   misc      :22
##  1st Qu.:583.5   1st Qu.: 30.75   1st Qu.:500   platform  :17
##  Median :603.9   Median : 60.50   Median :500   puzzle    :21
##  Mean   :603.8   Mean   : 60.50   Mean   :500   shooter   :22
##  3rd Qu.:621.7   3rd Qu.: 90.25   3rd Qu.:500   simulation:20
##  Max.   :688.5   Max.   :120.00   Max.   :500   strategy  :18
##        motivation          type          gender        income          age
##  competition:120   individual:  0   female:  0   Min.   :2068   Min.   :42.00
##  destruction:  0   social    :120   male  :120   1st Qu.:2795   1st Qu.:44.00
##                                                  Median :2975   Median :47.00
##                                                  Mean   :2985   Mean   :46.92
##                                                  3rd Qu.:3161   3rd Qu.:49.00
##                                                  Max.   :3869   Max.   :52.00
##  dummydecember
##  0:110
##  1: 10
##
##
##
##
```

```r
# Plot the spending of this gamer over 10 years
plot(df_Gamer500$spending, type = "l")
```



Comment on the characteristics of this gamer:

- This player is a male, one among 60% male players in 500 players here.
- He was 42 years old at the start of the research period and became 52 at the end. This means he was older than an average player in this data set.
- The motivation to play of this player was always for competition, which is similar to the majority of observations in the dataset.
- He always played games of social type, which is like the majority of observations in the dataset who likes to play social games.
- In terms of mean income, he earned 2985, 14.7% higher than the average of all players (2602).
- Regarding mean spending, he spent 603.8, 62.6% higher than the average of all players (371.3).
- Looking at the plot of spending, we can see the amount of money he spent on gaming fluctuated and did not stay the same for every month. There seems to be a seasonality of 12 months in his spending as well.
- Compared to player 1, player 500 had a completely opposite profile. He differs in gender, age cohort, has higher income, higher spending on game, plays different game type, and has different motivation to play.
- This is why basing the analysis on these two gamers is more beneficial.
- We can understand and determine which factors lead to changes in monthly spending on games for two different customer profiles instead of one, which gives us a bigger picture, better forecasting and more insights into what to do for each customer segment to boost spending.
- More importantly, by analyzing the 2 games, we can examine inter-dependency of their spendings and mutual effects of explanatory variables on the spendings at a given time. For example, there can be negative correlation between monthly spending of player 1 and player 500, which cannot be explored without analyzing the time series at the same time.

3B

```
# First, we need to check stationarity conditions for both 2 time series.
# Because VAR is an extension of AR model which also requires stationarity conditions.

# As we already check it for gamer 1, now we will check the stationarity of the spending data for gamer
adf.test(df_Gamer500$spending)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  df_Gamer500$spending
## Dickey-Fuller = -4.4699, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(df_Gamer500$spending)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  df_Gamer500$spending
## KPSS Level = 0.057396, Truncation lag parameter = 4, p-value = 0.1
```

Comment: - We reject Null Hypothesis in Unit Root Test (ADF) as the p-value = 0.01 < 0.05(H0: the time series is non-stationary). - Also, we cannot reject Null Hypothesis in KPSS Test as the p-value = 0.1 > 0.05 (H0: the time series is stationary). - Therefore, both of the tests support that the spending over time for gamer 500 is stationary.

- In conclusion, we do not need differencing calculation to apply VAR for these time series

```
# Second, estimate VAR(1) for the spendings of 2 gamers without explanatory variables
# Create a data frame that has variables of 2 gamers over time
df_2gamers <- df_Gamer1 %>%
  inner_join(df_Gamer500, by = "monthindex", suffix = c("1", "500"))

colnames(df_2gamers)[which(names(df_2gamers) == "dummydecember1")] <- "dummydecember"

# Make it time series data frame
ts_2gamers <- ts(df_2gamers)
```

```
library(vars)
```

```
## Warning: package 'vars' was built under R version 4.1.3

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 4.1.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.1.2

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.1.3

## Loading required package: urca
```

```
## Warning: package 'urca' was built under R version 4.1.3

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 4.1.3
```

```
# VAR(1) model without any explanatory variables:
# The reason "none" is chosen instead of "const" is because "const" leads to very negative r-square.
model3 <- VAR(ts_2gamers[, c("spending1", "spending500")], p=1, type="none")
summary(model3)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: spending1, spending500
## Deterministic variables: none
## Sample size: 119
## Log Likelihood: -1153.148
## Roots of the characteristic polynomial:
## 0.998 0.1002
## Call:
## VAR(y = ts_2gamers[, c("spending1", "spending500")], p = 1, type = "none")
##
##
## Estimation results for equation spending1:
## ==========================================
## spending1 = spending1.l1 + spending500.l1
##
##                Estimate Std. Error t value Pr(>|t|)
## spending1.l1   -0.09890    0.10284  -0.962    0.338
## spending500.l1  0.23998    0.02284  10.505   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 26.14 on 117 degrees of freedom
## Multiple R-Squared: 0.9629,  Adjusted R-squared: 0.9622
## F-statistic:  1517 on 2 and 117 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation spending500:
## ============================================
## spending500 = spending1.l1 + spending500.l1
##
##                Estimate Std. Error t value Pr(>|t|)
## spending1.l1   0.005988   0.161169   0.037     0.97
## spending500.l1 0.996657   0.035800  27.839   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 40.96 on 117 degrees of freedom
## Multiple R-Squared: 0.9955,  Adjusted R-squared: 0.9954
```

```
## F-statistic: 1.29e+04 on 2 and 117 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##             spending1 spending500
## spending1       682.8         468
## spending500     468.0        1676
##
## Correlation matrix of residuals:
##             spending1 spending500
## spending1      1.0000      0.4376
## spending500    0.4376      1.0000
```

Comment on the estimation results:

- Lagged (1) spending of gamer 500 plays an important role in predicting both players' spending at time t.
- Lagged (1) spending of gamer 500 has a statistically significant (p-value ~ 0) positive correlation with spending of gamer 1 at time t, with a fair estimate (0.24).
- However, lagged (1) spending of gamer 1 cannot help predict her spending at time t.
- Adjusted R-Square of the model is 0.9622, which is very high.
- Lagged (1) spending of gamer 500 has a statistically significant (p-value ~ 0) positive correlation with his spending at time t, with a very big estimate (0.997).
- Lagged (1) spending of gamer 1 cannot help predict the spending of gamer 500 at time t.
- Adjusted R-Square of the model is 0.9954, which is very high.

3C

```r
# VAR(1) with explanatory variables:
# We will include genre1, genre500, income1, income500, age1, age500, dummydecember
model5 <- VAR(ts_2gamers[, c("spending1", "spending500", "genre1", "genre500", "income1",
                              "income500", "age1", "age500", "dummydecember")], p=1, type="none")
summary(model5)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: spending1, spending500, genre1, genre500, income1, income500, age1, age500, du
## Deterministic variables: none
## Sample size: 119
## Log Likelihood: -3198.342
## Roots of the characteristic polynomial:
## 1.001 0.9783 0.2993 0.2993 0.2281 0.1772 0.1772 0.09365 0.04576
## Call:
## VAR(y = ts_2gamers[, c("spending1", "spending500", "genre1",
##     "genre500", "income1", "income500", "age1", "age500", "dummydecember")],
```

```
##     p = 1, type = "none")
##
##
## Estimation results for equation spending1:
## ==========================================
## spending1 = spending1.l1 + spending500.l1 + genre1.l1 + genre500.l1 + income1.l1 + income500.l1 + age
##
##                    Estimate Std. Error t value Pr(>|t|)
## spending1.l1       0.177623   0.124608   1.425  0.15686
## spending500.l1     0.074063   0.098684   0.751  0.45455
## genre1.l1          2.032616   1.311661   1.550  0.12410
## genre500.l1       -2.473642   1.354739  -1.826  0.07058 .
## income1.l1        -0.005208   0.007974  -0.653  0.51503
## income500.l1      -0.006622   0.007306  -0.906  0.36670
## age1.l1           -5.766838   2.961451  -1.947  0.05405 .
## age500.l1          5.848224   2.636587   2.218  0.02860 *
## dummydecember.l1 -38.377065  12.418822  -3.090  0.00253 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 23.73 on 110 degrees of freedom
## Multiple R-Squared: 0.9712,  Adjusted R-squared: 0.9689
## F-statistic: 412.4 on 9 and 110 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation spending500:
## ============================================
## spending500 = spending1.l1 + spending500.l1 + genre1.l1 + genre500.l1 + income1.l1 + income500.l1 + a
##
##                    Estimate Std. Error t value Pr(>|t|)
## spending1.l1       0.194183   0.155400   1.250   0.2141
## spending500.l1     0.109438   0.123070   0.889   0.3758
## genre1.l1          0.525117   1.635782   0.321   0.7488
## genre500.l1       -3.391547   1.689506  -2.007   0.0472 *
## income1.l1         0.001292   0.009944   0.130   0.8969
## income500.l1       0.000948   0.009111   0.104   0.9173
## age1.l1          -23.161980   3.693248  -6.271 7.21e-09 ***
## age500.l1         23.460232   3.288107   7.135 1.09e-10 ***
## dummydecember.l1 -29.863804  15.487607  -1.928   0.0564 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 29.6 on 110 degrees of freedom
## Multiple R-Squared: 0.9978,  Adjusted R-squared: 0.9976
## F-statistic:  5503 on 9 and 110 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation genre1:
## =======================================
## genre1 = spending1.l1 + spending500.l1 + genre1.l1 + genre500.l1 + income1.l1 + income500.l1 + age1.l
##
##                    Estimate Std. Error t value Pr(>|t|)
```

```
## spending1.l1     -0.0118164  0.0094543  -1.250   0.2140
## spending500.l1   -0.0041387  0.0074874  -0.553   0.5816
## genre1.l1        -0.2209388  0.0995192  -2.220   0.0285 *
## genre500.l1      -0.0442445  0.1027877  -0.430   0.6677
## income1.l1        0.0010713  0.0006050   1.771   0.0794 .
## income500.l1      0.0002631  0.0005543   0.475   0.6360
## age1.l1          -0.0022819  0.2246933  -0.010   0.9919
## age500.l1         0.0850464  0.2000449   0.425   0.6716
## dummydecember.l1  1.4619645  0.9422495   1.552   0.1236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.801 on 110 degrees of freedom
## Multiple R-Squared: 0.8017,  Adjusted R-squared: 0.7854
## F-statistic:  49.4 on 9 and 110 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation genre500:
## ==========================================
## genre500 = spending1.l1 + spending500.l1 + genre1.l1 + genre500.l1 + income1.l1 + income500.l1 + age
##
##                    Estimate Std. Error t value Pr(>|t|)
## spending1.l1     -3.350e-03  9.004e-03  -0.372   0.7105
## spending500.l1    5.254e-03  7.131e-03   0.737   0.4628
## genre1.l1        -6.796e-03  9.478e-02  -0.072   0.9430
## genre500.l1       4.582e-02  9.789e-02   0.468   0.6407
## income1.l1        3.166e-05  5.761e-04   0.055   0.9563
## income500.l1      9.998e-04  5.279e-04   1.894   0.0609 .
## age1.l1           9.624e-02  2.140e-01   0.450   0.6538
## age500.l1        -8.333e-02  1.905e-01  -0.437   0.6627
## dummydecember.l1 -7.875e-01  8.973e-01  -0.878   0.3821
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.715 on 110 degrees of freedom
## Multiple R-Squared: 0.8156,  Adjusted R-squared: 0.8005
## F-statistic: 54.06 on 9 and 110 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation income1:
## ========================================
## income1 = spending1.l1 + spending500.l1 + genre1.l1 + genre500.l1 + income1.l1 + income500.l1 + age1
##
##                   Estimate Std. Error t value Pr(>|t|)
## spending1.l1      -0.96451    1.47394  -0.654  0.51424
## spending500.l1    -0.30938    1.16730  -0.265  0.79148
## genre1.l1        -30.64128   15.51517  -1.975  0.05078 .
## genre500.l1      -17.25369   16.02473  -1.077  0.28397
## income1.l1         0.02768    0.09432   0.293  0.76970
## income500.l1       0.06222    0.08642   0.720  0.47308
## age1.l1          -71.20789   35.02995  -2.033  0.04448 *
## age500.l1         84.16506   31.18724   2.699  0.00806 **
```

```
## dummydecember.l1 -47.19896  146.89781  -0.321  0.74859
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 280.7 on 110 degrees of freedom
## Multiple R-Squared: 0.9815,  Adjusted R-squared: 0.9799
## F-statistic: 647.2 on 9 and 110 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation income500:
## ==========================================
## income500 = spending1.l1 + spending500.l1 + genre1.l1 + genre500.l1 + income1.l1 + income500.l1 + age
##
##                    Estimate Std. Error t value Pr(>|t|)
## spending1.l1        0.54693    1.60565   0.341   0.7340
## spending500.l1      2.33643    1.27160   1.837   0.0689 .
## genre1.l1         -14.96476   16.90156  -0.885   0.3779
## genre500.l1         1.14667   17.45665   0.066   0.9477
## income1.l1         -0.04687    0.10275  -0.456   0.6492
## income500.l1        0.20521    0.09414   2.180   0.0314 *
## age1.l1           -51.99843   38.16012  -1.363   0.1758
## age500.l1          54.70516   33.97404   1.610   0.1102
## dummydecember.l1 -274.93040  160.02417  -1.718   0.0886 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 305.8 on 110 degrees of freedom
## Multiple R-Squared: 0.9904,  Adjusted R-squared: 0.9896
## F-statistic:  1263 on 9 and 110 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation age1:
## =====================================
## age1 = spending1.l1 + spending500.l1 + genre1.l1 + genre500.l1 + income1.l1 + income500.l1 + age1.l1
##
##                    Estimate Std. Error t value Pr(>|t|)
## spending1.l1     -1.727e-03  1.406e-03  -1.228 0.221936
## spending500.l1   -2.707e-03  1.113e-03  -2.431 0.016660 *
## genre1.l1        -1.625e-02  1.480e-02  -1.098 0.274544
## genre500.l1       1.370e-02  1.528e-02   0.896 0.372077
## income1.l1       -5.192e-05  8.995e-05  -0.577 0.564999
## income500.l1     -8.212e-05  8.242e-05  -0.996 0.321251
## age1.l1           8.831e-01  3.341e-02  26.433  < 2e-16 ***
## age500.l1         1.055e-01  2.974e-02   3.547 0.000573 ***
## dummydecember.l1  1.289e-01  1.401e-01   0.920 0.359583
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.2677 on 110 degrees of freedom
## Multiple R-Squared: 0.9999,  Adjusted R-squared: 0.9999
## F-statistic: 1.067e+05 on 9 and 110 DF,  p-value: < 2.2e-16
```

```
##
##
## Estimation results for equation age500:
## =======================================
## age500 = spending1.l1 + spending500.l1 + genre1.l1 + genre500.l1 + income1.l1 + income500.l1 + age1.l
##
##                    Estimate Std. Error t value Pr(>|t|)
## spending1.l1      1.126e-03  1.491e-03   0.755    0.452
## spending500.l1   -2.427e-04  1.181e-03  -0.206    0.838
## genre1.l1         1.566e-02  1.569e-02   0.998    0.321
## genre500.l1      -5.172e-03  1.621e-02  -0.319    0.750
## income1.l1        3.394e-05  9.541e-05   0.356    0.723
## income500.l1      1.019e-04  8.742e-05   1.166    0.246
## age1.l1           2.633e-03  3.544e-02   0.074    0.941
## age500.l1         9.946e-01  3.155e-02  31.527   <2e-16 ***
## dummydecember.l1 -1.257e-01  1.486e-01  -0.846    0.399
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.284 on 110 degrees of freedom
## Multiple R-Squared:     1,   Adjusted R-squared:     1
## F-statistic: 3.629e+05 on 9 and 110 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation dummydecember:
## ==============================================
## dummydecember = spending1.l1 + spending500.l1 + genre1.l1 + genre500.l1 + income1.l1 + income500.l1 +
##
##                    Estimate Std. Error t value Pr(>|t|)
## spending1.l1      3.675e-03  1.460e-03   2.517   0.0133 *
## spending500.l1    2.774e-04  1.156e-03   0.240   0.8108
## genre1.l1         8.912e-03  1.537e-02   0.580   0.5631
## genre500.l1      -2.298e-02  1.587e-02  -1.448   0.1505
## income1.l1       -5.308e-05  9.342e-05  -0.568   0.5710
## income500.l1     -2.259e-05  8.559e-05  -0.264   0.7924
## age1.l1          -2.016e-02  3.470e-02  -0.581   0.5625
## age500.l1         3.095e-02  3.089e-02   1.002   0.3185
## dummydecember.l1 -2.973e-01  1.455e-01  -2.044   0.0434 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.278 on 110 degrees of freedom
## Multiple R-Squared: 0.9429,  Adjusted R-squared: 0.9383
## F-statistic: 201.9 on 9 and 110 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##               spending1 spending500     genre1  genre500    income1 income500
## spending1      563.1660    167.2333 -13.822597   3.53694   829.1204    40.062
## spending500    167.2333    875.8774  -7.136110 -12.29541   367.6223  1909.439
## genre1         -13.8226     -7.1361   3.241984   0.09370   -78.2469    29.778
```

```
## genre500         3.5369    -12.2954   0.093696   2.94035     -2.9582   -24.966
## income1        829.1204    367.6223 -78.246904  -2.95817 78796.9781  3219.488
## income500       40.0622   1909.4390  29.777967 -24.96585  3219.4880 93508.225
## age1            -1.6652     -2.3094   0.036813  -0.02000    -6.3952    -8.869
## age500          -0.4125     -0.3163  -0.007877   0.07537    -0.3107    -4.647
## dummydecember    4.0148      4.5815  -0.047616  -0.02333    -0.2489     8.268
##                    age1    age500 dummydecember
## spending1     -1.665198 -0.412529      4.014831
## spending500   -2.309383 -0.316255      4.581457
## genre1         0.036813 -0.007877     -0.047616
## genre500      -0.019997  0.075370     -0.023331
## income1       -6.395192 -0.310702     -0.248858
## income500     -8.868867 -4.647186      8.268484
## age1           0.071665 -0.006824     -0.009019
## age500        -0.006824  0.080632     -0.008481
## dummydecember -0.009019 -0.008481      0.077301
##
## Correlation matrix of residuals:
##               spending1 spending500   genre1   genre500   income1 income500
## spending1      1.000000     0.23811 -0.32349  0.086918  0.124464  0.005521
## spending500    0.238113     1.00000 -0.13392 -0.242282  0.044251  0.210989
## genre1        -0.323494    -0.13392  1.00000  0.030347 -0.154813  0.054083
## genre500       0.086918    -0.24228  0.03035  1.000000 -0.006146 -0.047613
## income1        0.124464     0.04425 -0.15481 -0.006146  1.000000  0.037507
## income500      0.005521     0.21099  0.05408 -0.047613  0.037507  1.000000
## age1          -0.262116    -0.29149  0.07637 -0.043562 -0.085103 -0.108340
## age500        -0.061218    -0.03763 -0.01541  0.154790 -0.003898 -0.053519
## dummydecember  0.608495     0.55679 -0.09512 -0.048937 -0.003189  0.097254
##                   age1    age500 dummydecember
## spending1     -0.26212 -0.061218      0.608495
## spending500   -0.29149 -0.037632      0.556788
## genre1         0.07637 -0.015407     -0.095115
## genre500      -0.04356  0.154790     -0.048937
## income1       -0.08510 -0.003898     -0.003189
## income500     -0.10834 -0.053519      0.097254
## age1           1.00000 -0.089764     -0.121180
## age500        -0.08976  1.000000     -0.107418
## dummydecember -0.12118 -0.107418      1.000000
```

Comment: - Lagged (1) age of gamer 500 plays an important role in predicting both players' spending at time t. It has positive influence on the spendings of gamer 1 and gamer 500. One unit change of it is associated with 23.46 change in spending of player 500 at significant level of 0.001, keeping other variables constant. One unit change of it is associated with 5.85 change in spending of player 1 at significant level of 0.05, keeping other variables constant. - Also, lagged (1) age of gamer 1 is negatively related to spending of gamer 500, at significant level of 0.001, with an estimate of -23.16. - For spending of player 500, his genre of games is negatively correlated with his spending, with an estimate of -3.39, at significant level of 0.05. - More importantly, lagged (1) spending of each gamer is not statistically associated with both players' spending at time t anymore. - Adjusted R-Square of the 2 models for spending 1 and spending 500 are 0.9689 and 0.9976, which are very high.

```r
# To compare this model to the one in question 3B, we will use BIC.
BIC(model3)
```

```
## [1] 2325.413
```

```
BIC(model5)
```

```
## [1] 6783.792
```

Comment: - VAR(1) model in question 3B uses fewer parameters to explain the changes in spending of 2 players, but still has high adjusted R-Square, so it has lower BIC. Therefore, it fits the data better.

3D

For Granger causality, we will examine significant variables in question 3B and 3C

```
# For model in question 3B
causality(model3, cause = "spending500")$Granger
```

```
##
##  Granger causality H0: spending500 do not Granger-cause spending1
##
## data:  VAR object model3
## F-Test = 110.36, df1 = 1, df2 = 234, p-value < 2.2e-16
```

Comment: p-value ~ 0 < 0.05 –> We reject H0. We conclude that spending500 Granger-causes spending1.

```
# For model in question 3C
causality(model5, cause = "dummydecember")$Granger
```

```
##
##  Granger causality H0: dummydecember do not Granger-cause spending1
##  spending500 genre1 genre500 income1 income500 age1 age500
##
## data:  VAR object model5
## F-Test = 1.974, df1 = 8, df2 = 990, p-value = 0.04664
```

```
causality(model5, cause = "age1")$Granger
```

```
##
##  Granger causality H0: age1 do not Granger-cause spending1 spending500
##  genre1 genre500 income1 income500 age500 dummydecember
##
## data:  VAR object model5
## F-Test = 8.5756, df1 = 8, df2 = 990, p-value = 2.4e-11
```

```
causality(model5, cause = "age500")$Granger
```

```
##
##  Granger causality H0: age500 do not Granger-cause spending1
##  spending500 genre1 genre500 income1 income500 age1 dummydecember
##
## data:  VAR object model5
## F-Test = 18.524, df1 = 8, df2 = 990, p-value < 2.2e-16
```

```r
causality(model5, cause = "genre500")$Granger
```

```
##
##  Granger causality H0: genre500 do not Granger-cause spending1
##  spending500 genre1 income1 income500 age1 age500 dummydecember
##
## data:  VAR object model5
## F-Test = 1.1669, df1 = 8, df2 = 990, p-value = 0.3161
```

Comment: - For dummydecember, p-value = 0.047 < 0.05. We reject H0. We conclude that dummydecember Granger-causes the changes of the other variables. - For age1, p-value ~ 0 < 0.05. We reject H0. We conclude that age1 Granger-causes the changes of the other variables. - For age500, p-value ~ 0 < 0.05. We reject H0. We conclude that age500 Granger-causes the changes of the other variables. - For genre500, p-value = 0.3161 > 0.05. We cannot reject H0. We conclude that genre does not Granger-cause the changes of the other variables.

```r
# To see if these gamers potentially belong to a gamer group where they play the same games
# We can apply Chi Square test
chisq.test(df_2gamers$genre1, df_2gamers$genre500, correct = FALSE)
```

```
## Warning in chisq.test(df_2gamers$genre1, df_2gamers$genre500, correct = FALSE):
## Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  df_2gamers$genre1 and df_2gamers$genre500
## X-squared = 38.441, df = 25, p-value = 0.04187
```

Comment: Because p-value is 0.0419 < 0.05, so we reject null hypothesis (The two variables are independent). We conclude that the 2 genres of games that player1 and player500 play are related to each other. They belong to the same group in terms of this.

3E

- The number of parameters to be estimated in model3 (question 3B) (lag 1, without intercept, without explanatory variables, without error terms) is: 2*2 = 4 (parameters).
- The number of parameters to be estimated in model5 (question 3C) (lag 1, without intercept, with explanatory variables, without error terms) is: 9*9 = 81 (parameters).

3F

Pros and cons of selecting 2 gamers as representative gamers in a VAR model instead of analyzing 500 gamers at the same time in a VAR model.

Pro: - 2 players belong to the same group where they play the same game patterns over time. Therefore, the spending of player 1 and spending of player 500 can be dependent on each other. This is not the case for all 500 players. So if we include all of them, we will likely have more noises. - It is less likely to be overfitting and less likely to suffer from the curse of dimensionality. When we include 500 players, we will have (4*500+1)*(4*500+1) = 4004001 parameters (lag 1 for spending, income, genre, age, dummydecember, without intercept). It is a rule of thumb that require minimum 10 observations per predictor, which is minimum number of 4004001*10 = 40040010 observations. Meanwhile, we only have 60000 observations for 500 players. - It is easier and more intuitive for interpretation than for 500 variables. - It is simpler for estimation. - It will require faster runtime.

Cons: - It does not take into account the effects of possible other players' spendings and explantory variables. It is high chance that the spending of player 1 does not just depend on player 500, also on some of the others of the same group.

QUESTION 4

4A

```
library(plm)
```

```
## Warning: package 'plm' was built under R version 4.1.3
```

```
##
## Attaching package: 'plm'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
```

```
library(graphics)
library(gplots)
```

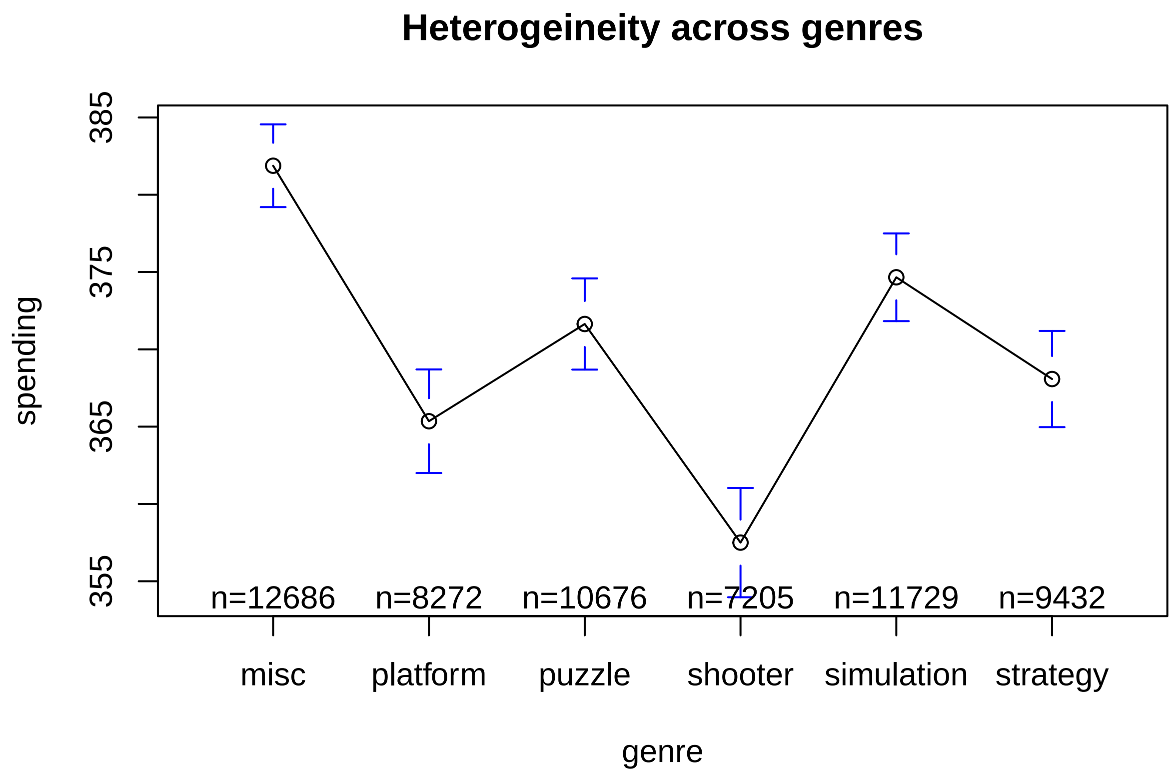
```
## Warning: package 'gplots' was built under R version 4.1.3
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
# create panel data index, make it a panel data frame
pn_Game <- pdata.frame(df_Game, index = c("playerindex", "monthindex"))
```

```
# Plot the spending data with respect to genres
plotmeans(spending ~ genre, data=pn_Game, main = "Heterogeineity across genres")
```



Comment: Spending differs between the genre of games that are played. It is on average the highest for misc genre, followed by simulation, puzzle, platform, strategy and finally shooter.

4B

```
# Individual fixed effects model
modelFixed1 <- plm(spending ~ income + genre +  age + dummydecember + motivation + type + gender
                   , data = pn_Game,
                    model = "within", effect = "individual")
print(summary(fixef(modelFixed1)))
```

```
##      Estimate Std. Error t-value  Pr(>|t|)
## 1   123.62175    1.00633 122.844 < 2.2e-16 ***
## 2   380.06264    1.08797 349.330 < 2.2e-16 ***
## 3   375.85181    1.08582 346.146 < 2.2e-16 ***
## 4   385.27524    1.10049 350.094 < 2.2e-16 ***
## 5   501.94376    1.01828 492.931 < 2.2e-16 ***
## 6   393.20750    1.09434 359.310 < 2.2e-16 ***
## 7    72.23850    1.03632  69.707 < 2.2e-16 ***
## 8   183.10010    1.05778 173.098 < 2.2e-16 ***
## 9   403.88444    1.01707 397.106 < 2.2e-16 ***
## 10  327.55294    1.02850 318.476 < 2.2e-16 ***
## 11  417.26601    1.04539 399.148 < 2.2e-16 ***
## 12  343.42209    1.01256 339.161 < 2.2e-16 ***
```

```
## 13  211.81733    1.15754 182.990 < 2.2e-16 ***
## 14  534.84787    1.07201 498.923 < 2.2e-16 ***
## 15  215.66109    1.13033 190.795 < 2.2e-16 ***
## 16  489.49981    1.15683 423.137 < 2.2e-16 ***
## 17  211.56986    1.07616 196.598 < 2.2e-16 ***
## 18  203.54936    1.05434 193.059 < 2.2e-16 ***
## 19  160.97292    1.12605 142.954 < 2.2e-16 ***
## 20  185.09482    1.04991 176.295 < 2.2e-16 ***
## 21  228.00255    1.14596 198.963 < 2.2e-16 ***
## 22  219.92316    1.14792 191.584 < 2.2e-16 ***
## 23  147.25734    1.02449 143.738 < 2.2e-16 ***
## 24   85.92760    1.02983  83.439 < 2.2e-16 ***
## 25  178.72992    1.06358 168.045 < 2.2e-16 ***
## 26  477.22279    1.02719 464.592 < 2.2e-16 ***
## 27  333.49725    1.12045 297.646 < 2.2e-16 ***
## 28  529.72980    1.06599 496.938 < 2.2e-16 ***
## 29  487.41318    1.18461 411.455 < 2.2e-16 ***
## 30   90.28471    1.18294  76.323 < 2.2e-16 ***
## 31  298.57317    1.02231 292.058 < 2.2e-16 ***
## 32  200.48849    1.09371 183.310 < 2.2e-16 ***
## 33  220.23615    1.13319 194.350 < 2.2e-16 ***
## 34  324.65164    1.00509 323.007 < 2.2e-16 ***
## 35  157.26662    1.01019 155.680 < 2.2e-16 ***
## 36  449.15902    1.17311 382.878 < 2.2e-16 ***
## 37  167.39883    1.02340 163.571 < 2.2e-16 ***
## 38  197.56267    1.15505 171.043 < 2.2e-16 ***
## 39  569.43185    1.08970 522.557 < 2.2e-16 ***
## 40  476.26051    1.03230 461.358 < 2.2e-16 ***
## 41  346.43634    1.08809 318.390 < 2.2e-16 ***
## 42  394.57798    1.05778 373.024 < 2.2e-16 ***
## 43  225.95137    1.07614 209.964 < 2.2e-16 ***
## 44  381.07066    1.00293 379.958 < 2.2e-16 ***
## 45  234.57368    1.08352 216.492 < 2.2e-16 ***
## 46  320.75669    1.06142 302.195 < 2.2e-16 ***
## 47  410.72356    1.06788 384.614 < 2.2e-16 ***
## 48  311.94159    1.03533 301.297 < 2.2e-16 ***
## 49  190.30626    1.17369 162.143 < 2.2e-16 ***
## 50  453.44473    1.09031 415.887 < 2.2e-16 ***
## 51  104.14873    1.00936 103.183 < 2.2e-16 ***
## 52  225.17231    1.04655 215.156 < 2.2e-16 ***
## 53  429.76033    1.01332 424.110 < 2.2e-16 ***
## 54  321.90864    1.08083 297.836 < 2.2e-16 ***
## 55  145.25188    1.15119 126.175 < 2.2e-16 ***
## 56  321.34770    1.15257 278.809 < 2.2e-16 ***
## 57  316.35326    1.18435 267.112 < 2.2e-16 ***
## 58  445.79895    1.11621 399.387 < 2.2e-16 ***
## 59  152.97830    0.99828 153.241 < 2.2e-16 ***
## 60  496.85153    1.08994 455.852 < 2.2e-16 ***
## 61  504.50341    1.00341 502.789 < 2.2e-16 ***
## 62   86.19714    1.12106  76.889 < 2.2e-16 ***
## 63  225.92636    1.03434 218.425 < 2.2e-16 ***
## 64   74.64011    1.15898  64.402 < 2.2e-16 ***
## 65  187.57478    1.16083 161.587 < 2.2e-16 ***
## 66  424.43220    1.07856 393.516 < 2.2e-16 ***
```

```
## 67  224.14421    1.06654 210.159 < 2.2e-16 ***
## 68  323.12213    1.08685 297.303 < 2.2e-16 ***
## 69   92.41226    1.04248  88.647 < 2.2e-16 ***
## 70  351.63594    1.09560 320.953 < 2.2e-16 ***
## 71  127.42857    1.08430 117.521 < 2.2e-16 ***
## 72  519.12338    1.09701 473.216 < 2.2e-16 ***
## 73   74.79600    1.01901  73.400 < 2.2e-16 ***
## 74  464.12144    1.17888 393.697 < 2.2e-16 ***
## 75  112.50135    1.10663 101.661 < 2.2e-16 ***
## 76  329.98698    1.14861 287.293 < 2.2e-16 ***
## 77  261.05862    1.01610 256.923 < 2.2e-16 ***
## 78  102.77786    1.03222  99.569 < 2.2e-16 ***
## 79  229.28153    1.00839 227.374 < 2.2e-16 ***
## 80  404.25096    1.14721 352.379 < 2.2e-16 ***
## 81  536.65532    1.05438 508.975 < 2.2e-16 ***
## 82  304.02230    1.17003 259.841 < 2.2e-16 ***
## 83  139.14804    1.10283 126.174 < 2.2e-16 ***
## 84  344.26776    1.04508 329.416 < 2.2e-16 ***
## 85  168.78229    1.01296 166.623 < 2.2e-16 ***
## 86  521.19394    1.17833 442.315 < 2.2e-16 ***
## 87  265.25338    1.03529 256.211 < 2.2e-16 ***
## 88  224.91290    1.10411 203.705 < 2.2e-16 ***
## 89  148.21224    1.06651 138.970 < 2.2e-16 ***
## 90  520.03369    1.02207 508.804 < 2.2e-16 ***
## 91  151.20729    1.02853 147.013 < 2.2e-16 ***
## 92  520.96810    1.10515 471.401 < 2.2e-16 ***
## 93  134.93275    1.03658 130.172 < 2.2e-16 ***
## 94  132.67476    1.04079 127.475 < 2.2e-16 ***
## 95  117.14057    1.01748 115.128 < 2.2e-16 ***
## 96  327.81345    1.10393 296.951 < 2.2e-16 ***
## 97  217.91202    1.10784 196.699 < 2.2e-16 ***
## 98   80.42515    1.06086  75.811 < 2.2e-16 ***
## 99  225.16707    1.17642 191.400 < 2.2e-16 ***
## 100 441.32517    1.15821 381.042 < 2.2e-16 ***
## 101  81.62353    1.04676  77.977 < 2.2e-16 ***
## 102 355.06659    1.04738 339.003 < 2.2e-16 ***
## 103 209.20774    1.13589 184.180 < 2.2e-16 ***
## 104 168.91579    1.03132 163.786 < 2.2e-16 ***
## 105 133.90521    1.04122 128.604 < 2.2e-16 ***
## 106 230.63243    1.05926 217.729 < 2.2e-16 ***
## 107 144.98536    1.05099 137.952 < 2.2e-16 ***
## 108 133.76683    1.16122 115.195 < 2.2e-16 ***
## 109 287.05396    1.18270 242.711 < 2.2e-16 ***
## 110  88.64115    1.14658  77.309 < 2.2e-16 ***
## 111 428.50402    1.02066 419.831 < 2.2e-16 ***
## 112 117.53602    1.08776 108.053 < 2.2e-16 ***
## 113 547.83600    1.05296 520.280 < 2.2e-16 ***
## 114 127.54468    1.13780 112.098 < 2.2e-16 ***
## 115 179.43825    1.12740 159.160 < 2.2e-16 ***
## 116 528.04762    1.10115 479.541 < 2.2e-16 ***
## 117 545.20411    1.16495 468.007 < 2.2e-16 ***
## 118 206.97046    1.02681 201.566 < 2.2e-16 ***
## 119 129.23487    1.02774 125.746 < 2.2e-16 ***
## 120 472.16038    1.00593 469.377 < 2.2e-16 ***
```

```
## 121 444.99304    1.03844 428.521 < 2.2e-16 ***
## 122 531.01799    1.02037 520.417 < 2.2e-16 ***
## 123 570.91577    1.10384 517.211 < 2.2e-16 ***
## 124 544.49273    1.02601 530.691 < 2.2e-16 ***
## 125 312.24781    1.02618 304.282 < 2.2e-16 ***
## 126 210.21703    1.03468 203.171 < 2.2e-16 ***
## 127 193.70802    1.14547 169.107 < 2.2e-16 ***
## 128 320.64113    1.02774 311.987 < 2.2e-16 ***
## 129 315.91341    1.09016 289.787 < 2.2e-16 ***
## 130 228.75155    1.02739 222.653 < 2.2e-16 ***
## 131 556.23832    1.04931 530.099 < 2.2e-16 ***
## 132 389.10523    1.08412 358.914 < 2.2e-16 ***
## 133 131.40439    1.03651 126.776 < 2.2e-16 ***
## 134 280.34256    1.03168 271.735 < 2.2e-16 ***
## 135 528.15101    1.16267 454.255 < 2.2e-16 ***
## 136 304.88091    1.09763 277.762 < 2.2e-16 ***
## 137 526.68264    1.06560 494.260 < 2.2e-16 ***
## 138 368.96503    1.11893 329.748 < 2.2e-16 ***
## 139 387.20705    1.09059 355.042 < 2.2e-16 ***
## 140 507.32382    1.09395 463.755 < 2.2e-16 ***
## 141 320.61984    1.00914 317.715 < 2.2e-16 ***
## 142 563.89162    1.08150 521.398 < 2.2e-16 ***
## 143 232.13566    1.00795 230.304 < 2.2e-16 ***
## 144 311.83329    1.03716 300.662 < 2.2e-16 ***
## 145 247.34898    1.17599 210.333 < 2.2e-16 ***
## 146 383.89852    1.10399 347.739 < 2.2e-16 ***
## 147 443.53941    1.15949 382.531 < 2.2e-16 ***
## 148 353.53410    1.07715 328.213 < 2.2e-16 ***
## 149 564.63381    1.12776 500.669 < 2.2e-16 ***
## 150 358.04346    1.08530 329.904 < 2.2e-16 ***
## 151 288.07675    1.14991 250.522 < 2.2e-16 ***
## 152 182.99129    1.06977 171.056 < 2.2e-16 ***
## 153 110.28104    1.13822  96.889 < 2.2e-16 ***
## 154 497.02422    1.07434 462.631 < 2.2e-16 ***
## 155 187.38698    1.13491 165.111 < 2.2e-16 ***
## 156 568.52355    1.06388 534.386 < 2.2e-16 ***
## 157 371.45318    1.08221 343.235 < 2.2e-16 ***
## 158 571.60512    1.15119 496.535 < 2.2e-16 ***
## 159 256.85556    1.06607 240.937 < 2.2e-16 ***
## 160 349.53226    1.11755 312.768 < 2.2e-16 ***
## 161 281.21715    1.07001 262.817 < 2.2e-16 ***
## 162 358.79897    1.16093 309.062 < 2.2e-16 ***
## 163 285.59742    1.13561 251.493 < 2.2e-16 ***
## 164 181.95780    1.06660 170.597 < 2.2e-16 ***
## 165 109.43533    1.02745 106.512 < 2.2e-16 ***
## 166 392.14931    1.14868 341.392 < 2.2e-16 ***
## 167 284.54221    1.00725 282.493 < 2.2e-16 ***
## 168 102.34560    1.08862  94.014 < 2.2e-16 ***
## 169 471.75282    1.10391 427.347 < 2.2e-16 ***
## 170 231.32740    1.10649 209.065 < 2.2e-16 ***
## 171 450.80415    1.00703 447.656 < 2.2e-16 ***
## 172 362.50982    1.14071 317.793 < 2.2e-16 ***
## 173 426.52890    1.17684 362.435 < 2.2e-16 ***
## 174 281.35031    1.03392 272.120 < 2.2e-16 ***
```

```
## 175 239.02315    1.05106 227.412 < 2.2e-16 ***
## 176 450.26386    1.15562 389.630 < 2.2e-16 ***
## 177 281.87106    1.12567 250.402 < 2.2e-16 ***
## 178 351.55754    1.15163 305.269 < 2.2e-16 ***
## 179 125.47115    1.06263 118.076 < 2.2e-16 ***
## 180 219.28572    1.14606 191.339 < 2.2e-16 ***
## 181 308.56135    1.10734 278.651 < 2.2e-16 ***
## 182 239.67597    1.01066 237.147 < 2.2e-16 ***
## 183 369.56929    1.08247 341.414 < 2.2e-16 ***
## 184 106.23058    1.04256 101.894 < 2.2e-16 ***
## 185 550.45904    1.05774 520.412 < 2.2e-16 ***
## 186  76.53357    1.11722  68.504 < 2.2e-16 ***
## 187 490.64666    1.06528 460.581 < 2.2e-16 ***
## 188 386.97559    1.18122 327.606 < 2.2e-16 ***
## 189 226.24368    1.02761 220.166 < 2.2e-16 ***
## 190 443.83649    1.08833 407.816 < 2.2e-16 ***
## 191 391.16111    1.08425 360.767 < 2.2e-16 ***
## 192 569.54893    1.08948 522.770 < 2.2e-16 ***
## 193 132.70752    1.09775 120.891 < 2.2e-16 ***
## 194 513.99279    1.07386 478.640 < 2.2e-16 ***
## 195 475.77294    1.04284 456.229 < 2.2e-16 ***
## 196 483.17210    1.05745 456.921 < 2.2e-16 ***
## 197 490.37882    1.10203 444.977 < 2.2e-16 ***
## 198 438.77227    1.04392 420.311 < 2.2e-16 ***
## 199 561.95949    1.01674 552.705 < 2.2e-16 ***
## 200 389.17375    1.03397 376.387 < 2.2e-16 ***
## 201 642.84355    1.10498 581.769 < 2.2e-16 ***
## 202 547.01278    1.09965 497.443 < 2.2e-16 ***
## 203 239.97194    1.11830 214.586 < 2.2e-16 ***
## 204 544.85135    1.06657 510.846 < 2.2e-16 ***
## 205 217.37076    1.07227 202.721 < 2.2e-16 ***
## 206 477.18246    1.12004 426.041 < 2.2e-16 ***
## 207 480.32765    1.04759 458.509 < 2.2e-16 ***
## 208 164.33143    1.17997 139.268 < 2.2e-16 ***
## 209 212.68206    1.13006 188.203 < 2.2e-16 ***
## 210 601.50269    1.16960 514.281 < 2.2e-16 ***
## 211 389.65362    1.08612 358.756 < 2.2e-16 ***
## 212 195.52843    1.17363 166.602 < 2.2e-16 ***
## 213 347.45883    1.14064 304.618 < 2.2e-16 ***
## 214 599.07433    1.13513 527.759 < 2.2e-16 ***
## 215 487.68132    1.12434 433.751 < 2.2e-16 ***
## 216 608.24136    1.07446 566.093 < 2.2e-16 ***
## 217 197.96562    1.21563 162.850 < 2.2e-16 ***
## 218 569.20274    1.15015 494.894 < 2.2e-16 ***
## 219 244.84733    1.15789 211.460 < 2.2e-16 ***
## 220 379.13168    1.21485 312.080 < 2.2e-16 ***
## 221 150.35195    1.21581 123.664 < 2.2e-16 ***
## 222 280.92632    1.12458 249.806 < 2.2e-16 ***
## 223 603.04772    1.14617 526.141 < 2.2e-16 ***
## 224 164.32944    1.13383 144.933 < 2.2e-16 ***
## 225 463.41921    1.19965 386.296 < 2.2e-16 ***
## 226 499.52075    1.05835 471.981 < 2.2e-16 ***
## 227 328.67797    1.06307 309.179 < 2.2e-16 ***
## 228 374.74803    1.05863 353.992 < 2.2e-16 ***
```

```
## 229 615.44580    1.15570 532.531 < 2.2e-16 ***
## 230 528.03191    1.08229 487.886 < 2.2e-16 ***
## 231 371.86352    1.12313 331.095 < 2.2e-16 ***
## 232 326.59194    1.10039 296.797 < 2.2e-16 ***
## 233 160.55558    1.16130 138.255 < 2.2e-16 ***
## 234 268.00569    1.15394 232.253 < 2.2e-16 ***
## 235 348.38395    1.11336 312.913 < 2.2e-16 ***
## 236 563.47437    1.16866 482.156 < 2.2e-16 ***
## 237 457.34833    1.10352 414.444 < 2.2e-16 ***
## 238 539.60894    1.06925 504.661 < 2.2e-16 ***
## 239 254.64578    1.09883 231.743 < 2.2e-16 ***
## 240 511.79912    1.14504 446.970 < 2.2e-16 ***
## 241 578.54470    1.11660 518.130 < 2.2e-16 ***
## 242 415.19393    1.04723 396.469 < 2.2e-16 ***
## 243 582.60490    1.17644 495.228 < 2.2e-16 ***
## 244 646.24374    1.16405 555.170 < 2.2e-16 ***
## 245 574.92384    1.14850 500.589 < 2.2e-16 ***
## 246 586.08268    1.15601 506.987 < 2.2e-16 ***
## 247 479.60362    1.08770 440.933 < 2.2e-16 ***
## 248 153.94391    1.12978 136.260 < 2.2e-16 ***
## 249 436.86321    1.07935 404.745 < 2.2e-16 ***
## 250 349.41931    1.05010 332.750 < 2.2e-16 ***
## 251 616.31985    1.06252 580.052 < 2.2e-16 ***
## 252 477.17105    1.11536 427.817 < 2.2e-16 ***
## 253 178.49829    1.08337 164.762 < 2.2e-16 ***
## 254 306.63020    1.14035 268.892 < 2.2e-16 ***
## 255 234.98388    1.11716 210.341 < 2.2e-16 ***
## 256 624.11275    1.22131 511.021 < 2.2e-16 ***
## 257 212.36166    1.09147 194.565 < 2.2e-16 ***
## 258 156.76834    1.14880 136.463 < 2.2e-16 ***
## 259 522.67714    1.19885 435.982 < 2.2e-16 ***
## 260 573.62568    1.18736 483.109 < 2.2e-16 ***
## 261 149.10779    1.05553 141.264 < 2.2e-16 ***
## 262 249.98798    1.08608 230.174 < 2.2e-16 ***
## 263 247.84779    1.10521 224.254 < 2.2e-16 ***
## 264 161.36750    1.12011 144.064 < 2.2e-16 ***
## 265 516.89385    1.11652 462.949 < 2.2e-16 ***
## 266 545.10922    1.20421 452.671 < 2.2e-16 ***
## 267 265.28682    1.06389 249.356 < 2.2e-16 ***
## 268 211.96303    1.14047 185.855 < 2.2e-16 ***
## 269 273.39659    1.11907 244.307 < 2.2e-16 ***
## 270 314.58352    1.19123 264.084 < 2.2e-16 ***
## 271 243.31444    1.21673 199.974 < 2.2e-16 ***
## 272 419.26281    1.18489 353.841 < 2.2e-16 ***
## 273 289.93266    1.06523 272.178 < 2.2e-16 ***
## 274 327.62869    1.06309 308.186 < 2.2e-16 ***
## 275 213.95883    1.09175 195.977 < 2.2e-16 ***
## 276 337.86036    1.15535 292.432 < 2.2e-16 ***
## 277 156.61919    1.19826 130.706 < 2.2e-16 ***
## 278 153.19755    1.05836 144.750 < 2.2e-16 ***
## 279 388.56364    1.14966 337.981 < 2.2e-16 ***
## 280 369.01742    1.07440 343.463 < 2.2e-16 ***
## 281 470.78417    1.11384 422.666 < 2.2e-16 ***
## 282 205.28063    1.05028 195.453 < 2.2e-16 ***
```

```
## 283 449.64270    1.19592 375.982 < 2.2e-16 ***
## 284 617.07898    1.08415 569.182 < 2.2e-16 ***
## 285 184.33914    1.11552 165.250 < 2.2e-16 ***
## 286 495.08802    1.06801 463.562 < 2.2e-16 ***
## 287 337.81160    1.15666 292.057 < 2.2e-16 ***
## 288 472.89805    1.17617 402.065 < 2.2e-16 ***
## 289 518.94262    1.05497 491.902 < 2.2e-16 ***
## 290 494.25322    1.05926 466.602 < 2.2e-16 ***
## 291 632.03204    1.13551 556.606 < 2.2e-16 ***
## 292 574.70925    1.10085 522.058 < 2.2e-16 ***
## 293 163.86307    1.06342 154.091 < 2.2e-16 ***
## 294 325.06771    1.10110 295.221 < 2.2e-16 ***
## 295 389.35577    1.05056 370.617 < 2.2e-16 ***
## 296 236.71442    1.12907 209.654 < 2.2e-16 ***
## 297 227.18839    1.07208 211.914 < 2.2e-16 ***
## 298 585.66595    1.16782 501.502 < 2.2e-16 ***
## 299 505.81042    1.14195 442.936 < 2.2e-16 ***
## 300 514.68795    1.04933 490.491 < 2.2e-16 ***
## 301 416.40316    1.14456 363.810 < 2.2e-16 ***
## 302 543.28220    1.05774 513.624 < 2.2e-16 ***
## 303 261.61488    1.20477 217.150 < 2.2e-16 ***
## 304 168.20331    1.15202 146.007 < 2.2e-16 ***
## 305 515.24372    1.12528 457.882 < 2.2e-16 ***
## 306 337.62276    1.19197 283.247 < 2.2e-16 ***
## 307 297.87896    1.05006 283.677 < 2.2e-16 ***
## 308 355.31050    1.14023 311.613 < 2.2e-16 ***
## 309 507.84872    1.15682 439.002 < 2.2e-16 ***
## 310 370.81543    1.19010 311.582 < 2.2e-16 ***
## 311 603.55605    1.21447 496.972 < 2.2e-16 ***
## 312 564.15023    1.14209 493.962 < 2.2e-16 ***
## 313 474.72367    1.14055 416.224 < 2.2e-16 ***
## 314 390.01367    1.12534 346.576 < 2.2e-16 ***
## 315 484.03672    1.14467 422.862 < 2.2e-16 ***
## 316 388.80607    1.07264 362.474 < 2.2e-16 ***
## 317 399.02984    1.10804 360.123 < 2.2e-16 ***
## 318 412.01299    1.09586 375.973 < 2.2e-16 ***
## 319 494.16455    1.21049 408.234 < 2.2e-16 ***
## 320 556.27975    1.08217 514.043 < 2.2e-16 ***
## 321 398.21565    1.13604 350.531 < 2.2e-16 ***
## 322 333.68622    1.12969 295.379 < 2.2e-16 ***
## 323 523.70741    1.05733 495.312 < 2.2e-16 ***
## 324 410.14495    1.10214 372.136 < 2.2e-16 ***
## 325 207.65604    1.08622 191.173 < 2.2e-16 ***
## 326 336.75390    1.11020 303.326 < 2.2e-16 ***
## 327 487.27758    1.15021 423.644 < 2.2e-16 ***
## 328 377.11478    1.17337 321.394 < 2.2e-16 ***
## 329 619.93399    1.10344 561.818 < 2.2e-16 ***
## 330 157.85723    1.06319 148.474 < 2.2e-16 ***
## 331 302.81676    1.20855 250.563 < 2.2e-16 ***
## 332 537.30861    1.17834 455.987 < 2.2e-16 ***
## 333 559.57451    1.20736 463.469 < 2.2e-16 ***
## 334 251.05517    1.04954 239.206 < 2.2e-16 ***
## 335 317.88153    1.21558 261.507 < 2.2e-16 ***
## 336 226.95463    1.05289 215.553 < 2.2e-16 ***
```

```
## 337 169.22126    1.15469 146.551 < 2.2e-16 ***
## 338 568.93499    1.14905 495.135 < 2.2e-16 ***
## 339 455.75456    1.10196 413.586 < 2.2e-16 ***
## 340 174.03458    1.17602 147.987 < 2.2e-16 ***
## 341 573.10280    1.06406 538.599 < 2.2e-16 ***
## 342 553.70389    1.09844 504.082 < 2.2e-16 ***
## 343 568.96878    1.20624 471.686 < 2.2e-16 ***
## 344 467.96882    1.09088 428.981 < 2.2e-16 ***
## 345 568.64184    1.16193 489.394 < 2.2e-16 ***
## 346 449.42767    1.11751 402.169 < 2.2e-16 ***
## 347 213.95242    1.07882 198.320 < 2.2e-16 ***
## 348 537.97991    1.19149 451.520 < 2.2e-16 ***
## 349 413.69093    1.19033 347.542 < 2.2e-16 ***
## 350 551.57194    1.06047 520.121 < 2.2e-16 ***
## 351 531.90199    1.12663 472.118 < 2.2e-16 ***
## 352 160.45479    1.19071 134.755 < 2.2e-16 ***
## 353 263.95951    1.09827 240.342 < 2.2e-16 ***
## 354 484.94569    1.06073 457.182 < 2.2e-16 ***
## 355 588.48960    1.05319 558.767 < 2.2e-16 ***
## 356 596.66343    1.09380 545.494 < 2.2e-16 ***
## 357 288.87177    1.15054 251.075 < 2.2e-16 ***
## 358 378.96082    1.09315 346.668 < 2.2e-16 ***
## 359 220.84999    1.17621 187.765 < 2.2e-16 ***
## 360 308.54769    1.14119 270.374 < 2.2e-16 ***
## 361 496.73675    1.20573 411.980 < 2.2e-16 ***
## 362 418.75883    1.16386 359.801 < 2.2e-16 ***
## 363 535.52674    1.10763 483.490 < 2.2e-16 ***
## 364 476.41385    1.12418 423.788 < 2.2e-16 ***
## 365 406.50861    1.14139 356.153 < 2.2e-16 ***
## 366 413.74477    1.14682 360.774 < 2.2e-16 ***
## 367 379.63836    1.05622 359.433 < 2.2e-16 ***
## 368 196.72923    1.06568 184.605 < 2.2e-16 ***
## 369 299.27867    1.17621 254.443 < 2.2e-16 ***
## 370 182.49508    1.05730 172.605 < 2.2e-16 ***
## 371 159.42896    1.13540 140.416 < 2.2e-16 ***
## 372 623.30014    1.07552 579.534 < 2.2e-16 ***
## 373 152.03978    1.08304 140.383 < 2.2e-16 ***
## 374 645.49060    1.07382 601.117 < 2.2e-16 ***
## 375 381.73199    1.06673 357.852 < 2.2e-16 ***
## 376 325.16956    1.19782 271.467 < 2.2e-16 ***
## 377 348.41373    1.16229 299.764 < 2.2e-16 ***
## 378 437.64537    1.07214 408.197 < 2.2e-16 ***
## 379 530.31088    1.06318 498.798 < 2.2e-16 ***
## 380 501.44974    1.19271 420.428 < 2.2e-16 ***
## 381 404.66690    1.09716 368.832 < 2.2e-16 ***
## 382 342.17709    1.09087 313.674 < 2.2e-16 ***
## 383 242.93839    1.05963 229.268 < 2.2e-16 ***
## 384 638.11319    1.11428 572.667 < 2.2e-16 ***
## 385 653.16438    1.08708 600.844 < 2.2e-16 ***
## 386 216.61615    1.18819 182.307 < 2.2e-16 ***
## 387 280.02932    1.14290 245.017 < 2.2e-16 ***
## 388 179.46270    1.07360 167.160 < 2.2e-16 ***
## 389 189.73980    1.10783 171.272 < 2.2e-16 ***
## 390 489.57860    1.10886 441.514 < 2.2e-16 ***
```

```
## 391 413.53005    1.06468 388.406 < 2.2e-16 ***
## 392 178.53826    1.21609 146.814 < 2.2e-16 ***
## 393 617.73762    1.19289 517.851 < 2.2e-16 ***
## 394 231.20852    1.16197 198.979 < 2.2e-16 ***
## 395 547.06489    1.06170 515.270 < 2.2e-16 ***
## 396 633.88229    1.08646 583.440 < 2.2e-16 ***
## 397 373.68989    1.12675 331.653 < 2.2e-16 ***
## 398 509.16454    1.22181 416.731 < 2.2e-16 ***
## 399 197.55911    1.07517 183.747 < 2.2e-16 ***
## 400 201.17944    1.09972 182.937 < 2.2e-16 ***
## 401 153.25545    1.21211 126.437 < 2.2e-16 ***
## 402 365.26428    1.16003 314.875 < 2.2e-16 ***
## 403 576.70142    1.05175 548.325 < 2.2e-16 ***
## 404 250.20647    1.14403 218.707 < 2.2e-16 ***
## 405 583.10155    1.19186 489.238 < 2.2e-16 ***
## 406 339.00318    1.06798 317.423 < 2.2e-16 ***
## 407 351.94450    1.05439 333.790 < 2.2e-16 ***
## 408 320.06583    1.21072 264.360 < 2.2e-16 ***
## 409 426.37269    1.08137 394.290 < 2.2e-16 ***
## 410 209.76875    1.09395 191.754 < 2.2e-16 ***
## 411 441.58949    1.13700 388.380 < 2.2e-16 ***
## 412 402.07479    1.09119 368.475 < 2.2e-16 ***
## 413 301.81955    1.07008 282.053 < 2.2e-16 ***
## 414 154.93064    1.13654 136.318 < 2.2e-16 ***
## 415 495.71254    1.21530 407.893 < 2.2e-16 ***
## 416 182.78916    1.18167 154.687 < 2.2e-16 ***
## 417 396.16571    1.21522 326.003 < 2.2e-16 ***
## 418 519.95402    1.05747 491.697 < 2.2e-16 ***
## 419 531.80894    1.12342 473.385 < 2.2e-16 ***
## 420 636.01135    1.07636 590.893 < 2.2e-16 ***
## 421 483.53670    1.15264 419.505 < 2.2e-16 ***
## 422 515.71450    1.07080 481.617 < 2.2e-16 ***
## 423 267.30950    1.05060 254.435 < 2.2e-16 ***
## 424 481.88190    1.06992 450.391 < 2.2e-16 ***
## 425 612.12876    1.05378 580.889 < 2.2e-16 ***
## 426 360.62519    1.08300 332.987 < 2.2e-16 ***
## 427 407.10584    1.21972 333.771 < 2.2e-16 ***
## 428 382.25742    1.04611 365.408 < 2.2e-16 ***
## 429 574.21629    1.12821 508.964 < 2.2e-16 ***
## 430 475.00741    1.15985 409.543 < 2.2e-16 ***
## 431 174.79018    1.12513 155.351 < 2.2e-16 ***
## 432 278.66960    1.20014 232.197 < 2.2e-16 ***
## 433 525.11057    1.20086 437.278 < 2.2e-16 ***
## 434 430.38074    1.09780 392.039 < 2.2e-16 ***
## 435 312.49351    1.05645 295.796 < 2.2e-16 ***
## 436 316.12317    1.19189 265.229 < 2.2e-16 ***
## 437 548.61983    1.19853 457.744 < 2.2e-16 ***
## 438 358.58188    1.10292 325.120 < 2.2e-16 ***
## 439 366.73049    1.13239 323.856 < 2.2e-16 ***
## 440 159.33018    1.07313 148.473 < 2.2e-16 ***
## 441 163.95640    1.07725 152.198 < 2.2e-16 ***
## 442 193.88793    1.12484 172.369 < 2.2e-16 ***
## 443 521.88820    1.14485 455.858 < 2.2e-16 ***
## 444 489.38116    1.15888 422.289 < 2.2e-16 ***
```

```
## 445 187.64724    1.12946 166.139 < 2.2e-16 ***
## 446 564.94342    1.11180 508.135 < 2.2e-16 ***
## 447 385.68895    1.06531 362.045 < 2.2e-16 ***
## 448 493.41731    1.12460 438.750 < 2.2e-16 ***
## 449 201.50424    1.07216 187.942 < 2.2e-16 ***
## 450 225.49119    1.21850 185.056 < 2.2e-16 ***
## 451 414.30536    1.16026 357.079 < 2.2e-16 ***
## 452 250.98951    1.10970 226.178 < 2.2e-16 ***
## 453 629.03506    1.15318 545.480 < 2.2e-16 ***
## 454 456.46772    1.10844 411.812 < 2.2e-16 ***
## 455 233.59080    1.20259 194.239 < 2.2e-16 ***
## 456 448.16092    1.06714 419.963 < 2.2e-16 ***
## 457 433.24347    1.10701 391.362 < 2.2e-16 ***
## 458 343.24641    1.11186 308.713 < 2.2e-16 ***
## 459 188.61800    1.06124 177.734 < 2.2e-16 ***
## 460 223.89694    1.07563 208.155 < 2.2e-16 ***
## 461 558.49831    1.07870 517.752 < 2.2e-16 ***
## 462 276.55553    1.17603 235.161 < 2.2e-16 ***
## 463 616.37378    1.06703 577.656 < 2.2e-16 ***
## 464 469.54641    1.14451 410.261 < 2.2e-16 ***
## 465 443.35059    1.07067 414.087 < 2.2e-16 ***
## 466 524.81659    1.07352 488.873 < 2.2e-16 ***
## 467 179.70920    1.16685 154.012 < 2.2e-16 ***
## 468 211.27017    1.08873 194.052 < 2.2e-16 ***
## 469 230.44816    1.13191 203.593 < 2.2e-16 ***
## 470 568.80007    1.17025 486.050 < 2.2e-16 ***
## 471 336.78053    1.21305 277.631 < 2.2e-16 ***
## 472 479.95873    1.06495 450.686 < 2.2e-16 ***
## 473 203.18098    1.13784 178.568 < 2.2e-16 ***
## 474 398.08953    1.04716 380.160 < 2.2e-16 ***
## 475 647.67241    1.21720 532.101 < 2.2e-16 ***
## 476 572.21913    1.06370 537.953 < 2.2e-16 ***
## 477 461.23816    1.21291 380.275 < 2.2e-16 ***
## 478 621.49459    1.05218 590.674 < 2.2e-16 ***
## 479 441.89339    1.12599 392.449 < 2.2e-16 ***
## 480 342.23308    1.15849 295.414 < 2.2e-16 ***
## 481 384.38867    1.14019 337.128 < 2.2e-16 ***
## 482 522.64205    1.09798 476.002 < 2.2e-16 ***
## 483 559.94975    1.07903 518.936 < 2.2e-16 ***
## 484 425.46697    1.19811 355.116 < 2.2e-16 ***
## 485 157.56992    1.15405 136.536 < 2.2e-16 ***
## 486 221.21533    1.13904 194.212 < 2.2e-16 ***
## 487 642.14663    1.12488 570.860 < 2.2e-16 ***
## 488 445.73967    1.19384 373.365 < 2.2e-16 ***
## 489 458.15312    1.18893 385.348 < 2.2e-16 ***
## 490 337.33905    1.07829 312.847 < 2.2e-16 ***
## 491 187.33282    1.20100 155.981 < 2.2e-16 ***
## 492 442.06655    1.09406 404.060 < 2.2e-16 ***
## 493 174.71399    1.07335 162.774 < 2.2e-16 ***
## 494 157.17014    1.10125 142.720 < 2.2e-16 ***
## 495 401.69732    1.17153 342.882 < 2.2e-16 ***
## 496 299.40779    1.14785 260.842 < 2.2e-16 ***
## 497 196.61085    1.20206 163.561 < 2.2e-16 ***
## 498 593.16921    1.05373 562.924 < 2.2e-16 ***
```

```
## 499 410.37334    1.13790 360.639 < 2.2e-16 ***
## 500 591.38889    1.19482 494.962 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(modelFixed1)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = spending ~ income + genre + age + dummydecember +
##     motivation + type + gender, data = pn_Game, effect = "individual",
##     model = "within")
##
## Balanced Panel: n = 500, T = 120, N = 60000
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -40.192412  -6.649933  -0.043528   6.712114  38.529398
##
## Coefficients:
##                     Estimate  Std. Error   t-value  Pr(>|t|)
## income            9.7146e-03  1.3615e-04   71.3523 < 2.2e-16 ***
## genreplatform    -4.0124e+01  1.4266e-01 -281.2496 < 2.2e-16 ***
## genrepuzzle      -1.9964e+01  1.3140e-01 -151.9323 < 2.2e-16 ***
## genreshooter     -6.0030e+01  1.5078e-01 -398.1306 < 2.2e-16 ***
## genresimulation -9.8394e+00  1.2791e-01  -76.9225 < 2.2e-16 ***
## genrestrategy    -3.0073e+01  1.3654e-01 -220.2572 < 2.2e-16 ***
## age               1.1825e-01  1.4016e-02    8.4364 < 2.2e-16 ***
## dummydecember1    5.0264e+01  1.4712e-01  341.6602 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    38087000
## Residual Sum of Squares: 5887900
## R-Squared:      0.84541
## Adj. R-Squared: 0.84409
## F-statistic: 40667.3 on 8 and 59492 DF, p-value: < 2.22e-16
```

Comment: - Fixed effects of all variables are all significantly different from 0 because the F-test is significant with p-value = 2.22e-16, lower than 0.05. - All individuals have different intercepts.

Now, let us go further and isolate the effects of monthindex on spending.

```
# Monthindex fixed effects model
modelFixed2 <- plm(spending ~ income + genre +  age + dummydecember + motivation + type + gender
                   , data = pn_Game,
                    model = "within", effect = "time")

# Effect of monthindex on spending
summary(fixef(modelFixed2, type = "dmean"))
```

```
##     Estimate Std. Error t-value  Pr(>|t|)
```

```
## 1    -3.6653     8.2182 -0.4460    0.6556
## 2    -1.9698     8.1690 -0.2411    0.8095
## 3    -1.7299     8.1891 -0.2112    0.8327
## 4    -1.5103     8.1944 -0.1843    0.8538
## 5    -2.0730     8.1799 -0.2534    0.7999
## 6    -2.4412     8.2032 -0.2976    0.7660
## 7    -1.7812     8.2183 -0.2167    0.8284
## 8    -1.5385     8.2126 -0.1873    0.8514
## 9    -1.7120     8.2066 -0.2086    0.8348
## 10   -2.3095     8.2008 -0.2816    0.7782
## 11   -2.6924     8.2348 -0.3270    0.7437
## 12   47.7788     8.1841  5.8380 5.311e-09 ***
## 13   -1.9393     8.1891 -0.2368    0.8128
## 14   -1.5969     8.2192 -0.1943    0.8460
## 15   -2.0352     8.2115 -0.2479    0.8043
## 16   -1.9890     8.1996 -0.2426    0.8083
## 17   -2.4211     8.2004 -0.2952    0.7678
## 18   -1.9393     8.2078 -0.2363    0.8132
## 19   -2.7096     8.2230 -0.3295    0.7418
## 20   -3.0845     8.2321 -0.3747    0.7079
## 21   -2.3053     8.2275 -0.2802    0.7793
## 22   -2.2749     8.2384 -0.2761    0.7824
## 23   -2.5651     8.2093 -0.3125    0.7547
## 24   46.8237     8.1975  5.7119 1.122e-08 ***
## 25   -1.5168     8.2346 -0.1842    0.8539
## 26   -3.1399     8.2377 -0.3812    0.7031
## 27   -2.9102     8.2270 -0.3537    0.7235
## 28   -2.9532     8.2192 -0.3593    0.7194
## 29   -3.7484     8.2535 -0.4542    0.6497
## 30   -3.1193     8.2368 -0.3787    0.7049
## 31   -3.7329     8.2364 -0.4532    0.6504
## 32   -3.4930     8.2223 -0.4248    0.6710
## 33   -3.7037     8.2641 -0.4482    0.6540
## 34   -2.7590     8.2471 -0.3345    0.7380
## 35   -3.9902     8.2396 -0.4843    0.6282
## 36   47.3756     8.2606  5.7351 9.790e-09 ***
## 37   -2.1196     8.2526 -0.2568    0.7973
## 38   -3.1019     8.2588 -0.3756    0.7072
## 39   -2.0707     8.2561 -0.2508    0.8020
## 40   -3.6821     8.2353 -0.4471    0.6548
## 41   -3.4411     8.2604 -0.4166    0.6770
## 42   -3.2493     8.2475 -0.3940    0.6936
## 43   -3.5788     8.2607 -0.4332    0.6648
## 44   -2.7136     8.2783 -0.3278    0.7431
## 45   -3.0761     8.2856 -0.3713    0.7104
## 46   -4.3211     8.2787 -0.5220    0.6017
## 47   -3.4054     8.2626 -0.4121    0.6802
## 48   46.2557     8.3001  5.5729 2.516e-08 ***
## 49   -3.5604     8.2695 -0.4305    0.6668
## 50   -4.3411     8.2812 -0.5242    0.6001
## 51   -3.5517     8.2929 -0.4283    0.6684
## 52   -4.4804     8.2790 -0.5412    0.5884
## 53   -4.4599     8.2742 -0.5390    0.5899
## 54   -4.0215     8.2777 -0.4858    0.6271
```

```
## 55   -4.0694     8.2924 -0.4907    0.6236
## 56   -4.5406     8.2688 -0.5491    0.5829
## 57   -3.3875     8.2847 -0.4089    0.6826
## 58   -4.3980     8.2885 -0.5306    0.5957
## 59   -3.7453     8.2784 -0.4524    0.6510
## 60   46.6732     8.3246  5.6066 2.072e-08 ***
## 61   -4.2976     8.2917 -0.5183    0.6042
## 62   -4.5051     8.2965 -0.5430    0.5871
## 63   -4.3530     8.3348 -0.5223    0.6015
## 64   -4.0860     8.3123 -0.4916    0.6230
## 65   -3.6896     8.3231 -0.4433    0.6576
## 66   -3.8148     8.3231 -0.4583    0.6467
## 67   -5.0707     8.3210 -0.6094    0.5423
## 68   -5.1890     8.3304 -0.6229    0.5334
## 69   -4.3777     8.3326 -0.5254    0.5993
## 70   -4.7553     8.3427 -0.5700    0.5687
## 71   -3.5961     8.3362 -0.4314    0.6662
## 72   45.9785     8.3254  5.5227 3.352e-08 ***
## 73   -4.4700     8.3210 -0.5372    0.5911
## 74   -4.4072     8.3408 -0.5284    0.5972
## 75   -5.2559     8.3400 -0.6302    0.5286
## 76   -4.3239     8.3340 -0.5188    0.6039
## 77   -4.6516     8.3227 -0.5589    0.5762
## 78   -4.9729     8.3484 -0.5957    0.5514
## 79   -5.5197     8.3393 -0.6619    0.5080
## 80   -5.2033     8.3401 -0.6239    0.5327
## 81   -4.8395     8.3327 -0.5808    0.5614
## 82   -5.2647     8.3585 -0.6299    0.5288
## 83   -4.6148     8.3375 -0.5535    0.5799
## 84   44.6912     8.3502  5.3521 8.724e-08 ***
## 85   -5.1763     8.3482 -0.6201    0.5352
## 86   -5.3357     8.3631 -0.6380    0.5235
## 87   -4.5400     8.3776 -0.5419    0.5879
## 88   -5.8708     8.3656 -0.7018    0.4828
## 89   -5.6715     8.3619 -0.6783    0.4976
## 90   -5.8818     8.3554 -0.7039    0.4815
## 91   -5.7346     8.3608 -0.6859    0.4928
## 92   -5.7310     8.3559 -0.6859    0.4928
## 93   -5.6966     8.3727 -0.6804    0.4963
## 94   -6.0167     8.3667 -0.7191    0.4721
## 95   -5.2213     8.3635 -0.6243    0.5324
## 96   44.0030     8.3749  5.2542 1.492e-07 ***
## 97   -4.6855     8.3881 -0.5586    0.5764
## 98   -5.4917     8.3789 -0.6554    0.5122
## 99   -5.6111     8.3902 -0.6688    0.5036
## 100  -5.9709     8.3824 -0.7123    0.4763
## 101  -5.2345     8.3621 -0.6260    0.5313
## 102  -5.2696     8.3928 -0.6279    0.5301
## 103  -6.5162     8.3928 -0.7764    0.4375
## 104  -6.2764     8.3656 -0.7503    0.4531
## 105  -6.2754     8.4018 -0.7469    0.4551
## 106  -6.0316     8.3897 -0.7189    0.4722
## 107  -6.2004     8.4145 -0.7369    0.4612
## 108  44.3403     8.4130  5.2705 1.366e-07 ***
```

```
## 109  -5.7466     8.3895 -0.6850    0.4934
## 110  -6.2782     8.3938 -0.7480    0.4545
## 111  -6.7566     8.4103 -0.8034    0.4218
## 112  -6.0731     8.4292 -0.7205    0.4712
## 113  -6.7860     8.4163 -0.8063    0.4201
## 114  -6.4908     8.4005 -0.7727    0.4397
## 115  -6.6028     8.4132 -0.7848    0.4326
## 116  -6.5298     8.4102 -0.7764    0.4375
## 117  -6.3940     8.4094 -0.7603    0.4471
## 118  -6.6868     8.4164 -0.7945    0.4269
## 119  -5.7514     8.4091 -0.6839    0.4940
## 120  44.5399     8.3974  5.3040 1.137e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comment: - We can see that monthindex 12, 24, 36, 48, 60, 72, 84, 96, 108, 120 are significant at 0.001 level, which mean spending across individuals are higher than usual.

4C

- Similar to model in question 2, all the coefficients in model 4B are also statistically significant.
- Compared to the model in question 2, the estimates here show some differences, for example the estimate for age. Here is more generalized, and are the remainders after considering the dummy variable for each individual. The older the player is, the more he/she spends on gaming. In question 2, the model is only applied to player 1, so the older the player 1, the less she spends on gaming. The estimates differ in terms of interpretation.
- The first advantage of this panel data model to the representative individual model is that we can use this to predict spending for every single player rather than just the first player.
- Also, we can see 2 different types of effects: common effects shared between all players of explanatory variables on spending, and individual effects for each players.
- Last, we can also include variables like motivation, type of game, etc. in panel data model easier, which might not be the case for individual. For example, for player 1, we cannot include motivation, type, and gender because these variables are fixed for that player.

4D

```
# Estimate a random effects model
modelRandom1 <- plm(spending ~ income + genre +  age + dummydecember
                    + motivation + type + gender, data = pn_Game,
                   model = "random", effect = "individual", random.method = "amemiya")
print(summary(modelRandom1))
```

```
## Oneway (individual) effect Random Effect Model
##    (Amemiya's transformation)
##
## Call:
## plm(formula = spending ~ income + genre + age + dummydecember +
##     motivation + type + gender, data = pn_Game, effect = "individual",
```

```
##     model = "random", random.method = "amemiya")
##
## Balanced Panel: n = 500, T = 120, N = 60000
##
## Effects:
##                     var   std.dev share
## idiosyncratic    98.957     9.948 0.004
## individual    23419.228   153.033 0.996
## theta: 0.9941
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -41.311763  -6.704957  -0.024554   6.718881  38.983415
##
## Coefficients:
##                          Estimate  Std. Error   z-value  Pr(>|z|)
## (Intercept)            3.4476e+02  1.6161e+01   21.3336 < 2.2e-16 ***
## income                 9.7145e-03  1.3611e-04   71.3740 < 2.2e-16 ***
## genreplatform         -4.0124e+01  1.4262e-01 -281.3389 < 2.2e-16 ***
## genrepuzzle           -1.9964e+01  1.3136e-01 -151.9805 < 2.2e-16 ***
## genreshooter          -6.0030e+01  1.5073e-01 -398.2572 < 2.2e-16 ***
## genresimulation       -9.8394e+00  1.2787e-01  -76.9472 < 2.2e-16 ***
## genrestrategy         -3.0073e+01  1.3649e-01 -220.3271 < 2.2e-16 ***
## age                    1.1837e-01  1.4010e-02    8.4492 < 2.2e-16 ***
## dummydecember1         5.0264e+01  1.4707e-01  341.7686 < 2.2e-16 ***
## motivationdestruction -4.2773e+01  1.5817e+01   -2.7043 0.0068451 **
## typesocial            -1.0460e+01  1.6060e+01   -0.6513 0.5148405
## gendermale             6.6256e+01  1.7748e+01    3.7332 0.0001891 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Total Sum of Squares:    38136000
## Residual Sum of Squares: 5933200
## R-Squared:      0.84442
## Adj. R-Squared: 0.84439
## Chisq: 325587 on 11 DF, p-value: < 2.22e-16
```

```
ranef(modelRandom1)
```

```
##            1            2            3            4            5            6
## -178.3641856   78.0659171   84.3152985   50.9663377  199.9442044   48.4384369
##            7            8            9           10           11           12
## -219.2863210 -118.8890778  101.8883685  -17.2124972  125.7288762   51.8883882
##           13           14           15           16           17           18
##  -79.7146235  190.0742046  -75.8705468  144.7262577  -90.4207137 -130.7522205
##           19           20           21           22           23           24
## -141.0167676 -159.6660713  -73.9898038  -71.6089447 -154.7298735 -216.0576055
##           25           26           27           28           29           30
## -112.7991726  175.2239050  -11.2701308  227.7282065  185.4110750 -211.7033870
##           31           32           33           34           35           36
##  -46.1910839  -91.0419804  -81.7557308   -9.6531382 -144.7205467  147.1583971
##           37           38           39           40           41           42
## -177.3608039  -93.9687284  277.8884387  131.4897635   54.9008316  103.0413380
##           43           44           45           46           47           48
```

```
##  -65.5796393  79.0757318  -99.7295590  18.7625338  76.4143777   9.9482896
##           49          50           51          52          53          54
## -154.4570044 108.6735994 -197.8365652 -66.3580964 138.2234446  30.3741690
##           55          56           57          58          59          60
## -146.2776296  19.3519281  14.3571890 143.7994391 -149.0085078 194.8506318
##           61          62           63          64          65          66
##  159.7323819 -205.3297872 -76.0638499 -227.3470358 -114.4163969 122.4340691
##           67          68           69          70          71          72
## -120.6183896  21.1274602 -209.5734110   6.8682765 -174.5592144 174.3498058
##           73          74           75          76          77          78
## -227.1885517 129.8084633 -189.4862883  38.4510123 -83.7041635 -199.2079653
##           79          80           81          82          83          84
##  -62.2481347 102.2523540 234.6537618   2.0268615 -205.6122258  -0.4986474
##           85          86           87          88          89          90
## -175.9771542 219.1907358 -26.2782177 -66.6186296 -143.3159230 175.2616387
##           91          92           93          94          95          96
## -150.7801128 229.4261348 -199.3660581 -169.3123501 -184.8453819 -16.9535080
##           97          98           99         100         101         102
##  -84.0793822 -221.5604888 -76.8256464 139.3250782 -220.3618139  20.7598106
##          103         104          105         106         107         108
##  -82.3237694 -133.0723664 -210.8537255 -60.8984420 -146.5422930 -200.5342365
##          109         110          111         112         113         114
##  -14.9410952 -245.6580488  83.7352568 -173.9914057 256.2940824 -174.4440130
##          115         116          117         118         119         120
## -122.5520818 236.5054820 243.2002820 -95.0188667 -172.7517958 127.3904499
##          121         122          123         124         125         126
##  153.4550250 186.2455638 268.9120327 252.9514916  10.2547288 -81.3126335
##          127         128          129         130         131         132
## -108.2831704  18.6476811  13.9189126 -116.0103953 211.4643654  44.3364949
##          133         134          135         136         137         138
## -213.3542859 -11.1894521 226.1478193  13.3466805 224.6811907  66.9681686
##          139         140          141         142         143         144
##   52.8982633 205.3224740 -24.1448618 261.8885408 -59.3941279 -32.9317791
##          145         146          147         148         149         150
##  -44.1845249  81.9013443 151.9992615  61.9985446 230.3181565  66.5075984
##          151         152          153         154         155         156
##  -13.9178005 -161.7699190 -181.2470482 152.2518490 -146.9155216 234.2088873
##          157         158          159         160         161         162
##   69.4568579 226.8287586 -45.1364045  15.2243439 -20.7757878  67.2617861
##          163         164          165         166         167         168
##  -16.3967979 -109.5715095 -235.3223705  57.8393764  -6.9893455 -199.6413211
##          169         170          171         172         173         174
##  126.9807981 -113.4361937 106.0348791  70.9728156  81.7573007 -20.6418076
##          175         176          177         178         179         180
##  -62.9679086 148.2635215 -62.8946219  49.5607173 -176.5160645 -72.2462962
##          181         182          183         184         185         186
##   17.0268281 -62.3141393  67.5730476 -195.7555997 248.4568722 -225.4529622
##          187         188          189         190         191         192
##  188.6464557  95.4370648 -108.0581119  99.0656893  99.6241062 235.2337694
##          193         194          195         196         197         198
## -212.0524752 211.9916193 173.7737113 138.4006057 188.3779530 136.7743994
##          199         200          201         202         203         204
##  259.9578686  87.1778652 231.8130081 189.2175458 -160.5847133 144.2850869
##          205         206          207         208         209         210
```

```
## -183.1840390  119.3892267  122.5359868 -236.2237056 -145.1019988  200.9323812
##          211          212          213          214          215          216
##  -10.9076010 -215.4877104  -10.3302020  188.0446879   76.6558572  207.6727048
##          217          218          219          220          221          222
## -213.0512121  158.1738985 -112.9383939  -21.4315600 -250.2032312 -119.6318720
##          223          224          225          226          227          228
##  245.2495492 -236.2248406   52.3933106   98.9563584  -71.8805084   16.9597781
##          229          230          231          232          233          234
##  204.4151948  127.4659305   14.0739691  -73.9673398 -197.2272686 -132.5526200
##          235          236          237          238          239          240
##  -52.1763720  162.9053744   99.5561405  139.0429543 -113.5992164  111.2323872
##          241          242          243          244          245          246
##  177.9762191   57.4045659  182.0351533  235.2119607  174.3548712  185.5131586
##          247          248          249          250          251          252
##   68.5791391 -246.6099649   36.3004793  -61.5996065  215.7512518   76.6061170
##          253          254          255          256          257          258
## -222.0554505  -93.9291782 -122.8007611  266.3125948 -188.1934779 -243.7859626
##          259          260          261          262          263          264
##  122.1091062  173.0560485 -219.1324135 -150.5683247 -109.9369753 -196.4146000
##          265          266          267          268          269          270
##  105.8674326  144.5403041 -135.2694540 -188.5930766 -127.1611973  -85.9770444
##          271          272          273          274          275          276
## -114.4722765   18.6986632 -110.6245370  -30.1579516 -197.0563664  -73.1603998
##          277          278          279          280          281          282
## -243.9359401 -247.3545296  -11.9987772  -31.5427653   70.2194299 -205.7332062
##          283          284          285          286          287          288
##   49.0773204  216.5097895 -216.2154678  137.2953220  -62.7492321  115.1038861
##          289          290          291          292          293          294
##  118.3776594   93.6889226  231.4612504  174.1412743 -236.6896470  -75.4914982
##          295          296          297          298          299          300
##  -21.6645636 -131.5305060 -173.3667043  227.8680504   94.7839281  114.1233057
##          301          302          303          304          305          306
##    5.3797312  142.7162970  -96.1722312 -189.5796815  114.6772890  -62.9386215
##          307          308          309          310          311          312
## -102.6780421   -2.4787853  150.0536908  -29.7471129  202.9849022  163.5816907
##          313          314          315          316          317          318
##   74.1582952   32.2233708   73.0109397  -11.7548119  -11.9921758    0.9907171
##          319          320          321          322          323          324
##   93.5973166  145.2527509   -2.3468530  -24.1021782  123.1421802   -0.8773308
##          325          326          327          328          329          330
## -192.8988319  -63.8059475   86.7116526  -33.9076841  208.9042834 -242.6952427
##          331          332          333          334          335          336
## -108.2036679  136.7403529  201.7768319 -106.7284229  -93.1395676 -173.6000741
##          337          338          339          340          341          342
## -231.3335772  157.9061849   55.1906959 -226.5207668  172.5356323  153.1366076
##          343          344          345          346          347          348
##  168.3990109   67.4048077  210.8445922   38.4036291 -186.6024804  137.4114731
##          349          350          351          352          353          354
##   13.1268812  193.7773858  131.3349214 -197.3285412 -104.2857675   84.3817862
##          355          356          357          358          359          360
##  187.9222655  196.0947003 -111.6872037  -21.6001849 -147.3951867  -92.0117881
##          361          362          363          364          365          366
##   96.1695251   18.1950186  134.9599201   75.8487770    5.9456469   13.1815158
##          367          368          369          370          371          372
```

```
##   21.8500155 -161.0529185 -101.2811287 -218.0581781 -241.1251823  222.7308788
##          373          374          375          376          377          378
## -248.5129616  244.9206723  -18.8284125  -43.0797249  -52.1475384   79.8546166
##          379          380          381          382          383          384
##  129.7453727  100.8825809    4.1049052  -15.6107033 -157.6169828  280.3143803
##          385          386          387          388          389          390
##  242.1337685 -183.9409188 -120.5291610 -221.0908021 -210.8148679   89.0133818
##          391          392          393          394          395          396
##    2.5085121 -222.0179214  217.1663522 -169.3486062  146.4987201  233.3124665
##          397          398          399          400          401          402
##   15.9001546  108.5966379 -213.4551378 -199.3754708 -247.2997837  -35.2975280
##          403          404          405          406          407          408
##  208.4462238 -150.3509795  182.5314932  -61.5557750  -48.6145945  -80.4952465
##          409          410          411          412          413          414
##   25.8102706 -148.0144794   83.7971747   44.2848642  -98.7381290 -202.8515988
##          415          416          417          418          419          420
##   95.1451789 -228.2266554   -4.3981088  119.3889663  120.7818436  224.9816329
##          421          422          423          424          425          426
##   82.9708341  104.6891292 -133.2464113   81.3177629  254.3324030  -39.9348408
##          427          428          429          430          431          432
##   49.3133359  -18.3025353  173.6477090  117.2135068 -225.7643543  -89.5781314
##          433          434          435          436          437          438
##  124.5424277   29.8177904  -88.0642128  -84.4374453  148.0508956    0.7931365
##          439          440          441          442          443          444
##  -33.8309090 -241.2226596 -247.0567423 -206.6671845  164.0929567   78.3550089
##          445          446          447          448          449          450
## -212.9077374  153.9154306  -14.8715711   92.8516201 -199.0499736 -185.5266972
##          451          452          453          454          455          456
##   56.5135558 -149.5673611  228.4640852   98.6754460 -177.4271565   47.5981413
##          457          458          459          460          461          462
##   32.6802906  -57.3137036 -169.1636924 -133.8864296  147.4713188  -81.2316368
##          463          464          465          466          467          468
##  215.8049906   68.9811795   42.7878196  113.7909112 -231.3062209 -189.2849073
##          469          470          471          472          473          474
## -170.1083520  168.2308875  -31.4694253   79.3947990 -197.3747347  -12.9310163
##          475          476          477          478          479          480
##  247.0996714  171.6520372  103.4438502  220.9260352   41.3295191  -58.3279137
##          481          482          483          484          485          486
##  -26.6334938  122.0759058  148.9226609   24.9024386 -242.9845022 -179.3410503
##          487          488          489          490          491          492
##  241.5757074   45.1744690   57.5875426  -63.2201615 -170.4516847   41.5033123
##          493          494          495          496          497          498
## -225.8393980 -200.6113968   43.9057765  -68.8397436 -203.9457865  192.6016890
##          499          500
##    9.8103391  190.8184941
```

Comment: - In the case of 4B, we are accounting for fixed effects (or spending independent of time), while the model here is accounting for random effects (including time). - All individuals have different intercepts. - Effects of all variables here are all significantly different from 0 because the F-test is significant with p-value = 2.22e-16, lower than 0.0. However, looking at the p-value of typesocial, it is not significant as it is 0.5, larger than 0.05. - The random effects model here has "type", "motivation" and "gender" as explantory variables in addition to variables in the fixed effects model. - The adjusted R-square in both models are approximately the same.

4E

```r
# Use a Hausman test to test whether the additional exogeneity assumption is violated.
# H0 : Both fixed and random effects models can be used.
# (the exogeneity assumption in the random effects model is not violated).
# HA : One of the models is inconsistent.
# (the random effects model is inconsistent, since the fixed effects model does not have the
# exogeneity assumption).

phtest(modelFixed1, modelRandom1)
```

```
##
##  Hausman Test
##
## data:  spending ~ income + genre + age + dummydecember + motivation +  ...
## chisq = 0.092273, df = 8, p-value = 1
## alternative hypothesis: one model is inconsistent
```

Comment: - As the p-value is 1, higher than 0.05, we accept the H0. - The exogeneity assumption in the random effects model is not violated. The random effects model is consistent and can be used.

4F

There is an unaccounted property in the panel data estimation compared to the models in previous questions. It lacks lagged dependent and independent variables in the model. This can cause bias, because it does not really consider the impacts of lagged values on the values at time t.

QUESTION 5

5A

```r
# Create binary series
df_Game$largeSpending <- ifelse(df_Game$spending > 450, 1, 0)
```

```r
# Select the player with the equal number of high and low monthly spendings
# So that we have balanced category for dependent variable
library(dplyr)

df_Count <- df_Game %>%
  group_by(playerindex) %>%
  summarize(sum = sum(largeSpending))
df_Count %>% filter(sum == 60)
```

```
## # A tibble: 1 x 2
##   playerindex   sum
##         <int> <dbl>
## 1         121    60
```

We can choose player 121 as a representative player because that player has 60 months with high spendings and 60 months with low spendings.

5B

```
# Create data frame for player 121
df_Gamer121 <-  df_Game[which(df_Game$playerindex == 121), ]
df_Gamer121$largeSpending <- as.factor(df_Gamer121$largeSpending)
```

```
# Find starting points
beta0 <- lm(df_Gamer121$largeSpending ~ df_Gamer121$income + df_Gamer121$genre +
              df_Gamer121$dummydecember + df_Gamer121$age - 1)$coef

beta0
```

```
##          df_Gamer121$income       df_Gamer121$genremisc
##                0.0002343855                0.9126565283
##   df_Gamer121$genreplatform     df_Gamer121$genrepuzzle
##                0.0898715988                0.3842677041
##    df_Gamer121$genreshooter df_Gamer121$genresimulation
##                0.0783118974                0.6973421550
##   df_Gamer121$genrestrategy  df_Gamer121$dummydecember1
##                0.0449112421                0.3816111603
##             df_Gamer121$age
##                0.0171573070
```

```
# Use the glm function to estimate the logit model
modelLogit <- glm(largeSpending ~ income + genre + dummydecember + age - 1,
                  family=binomial(link='logit'), start = beta0, data = df_Gamer121)

logLik(modelLogit)
```

```
## 'log Lik.' -27.36017 (df=9)
```

```
modelLogit$coefficients
```

```
##          income       genremisc   genreplatform     genrepuzzle    genreshooter
##     0.003603852   -11.885204779   -36.171255056   -16.259706991   -35.764655659
## genresimulation   genrestrategy  dummydecember1             age
##   -14.429018601   -37.020777556    38.358595175     0.280187048
```

```
summary(modelLogit)
```

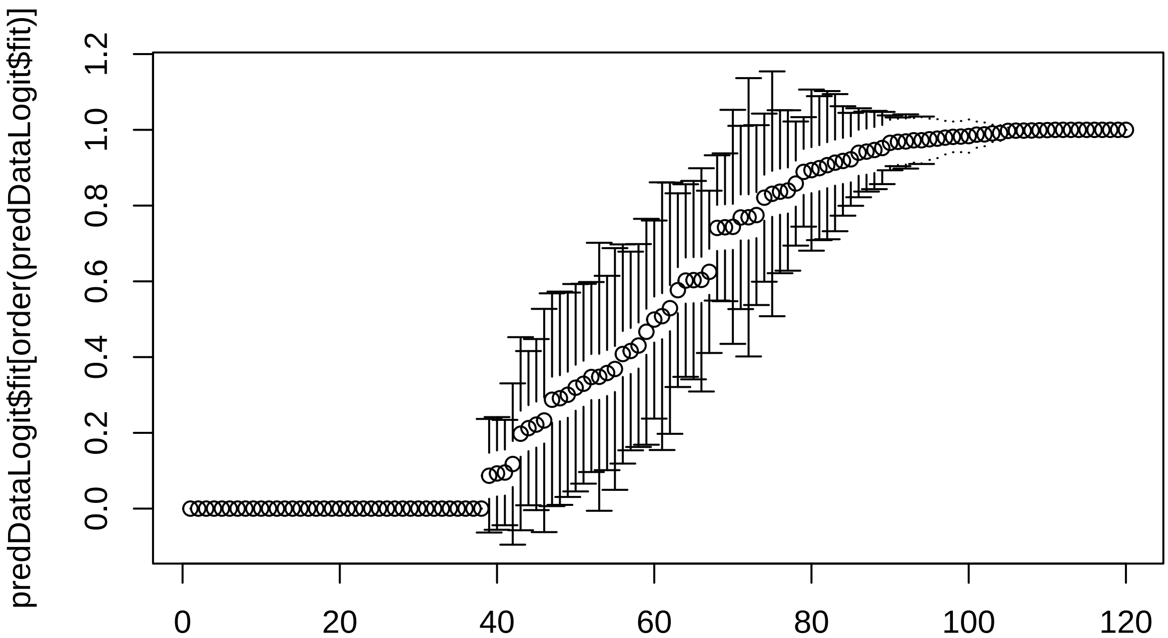
```
##
## Call:
## glm(formula = largeSpending ~ income + genre + dummydecember +
##     age - 1, family = binomial(link = "logit"), data = df_Gamer121,
##     start = beta0)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.17824  -0.00006   0.00000   0.25609   1.55050
##
```

```
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## income           3.604e-03  1.461e-03   2.467  0.01361 *
## genremisc       -1.189e+01  5.487e+00  -2.166  0.03030 *
## genreplatform   -3.617e+01  4.438e+03  -0.008  0.99350
## genrepuzzle     -1.626e+01  5.832e+00  -2.788  0.00530 **
## genreshooter    -3.576e+01  1.105e+04  -0.003  0.99742
## genresimulation -1.443e+01  5.496e+00  -2.625  0.00865 **
## genrestrategy   -3.702e+01  3.339e+03  -0.011  0.99115
## dummydecember1   3.836e+01  5.852e+03   0.007  0.99477
## age              2.802e-01  1.352e-01   2.072  0.03824 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 166.36  on 120  degrees of freedom
## Residual deviance:  54.72  on 111  degrees of freedom
## AIC: 72.72
##
## Number of Fisher Scoring iterations: 19
```

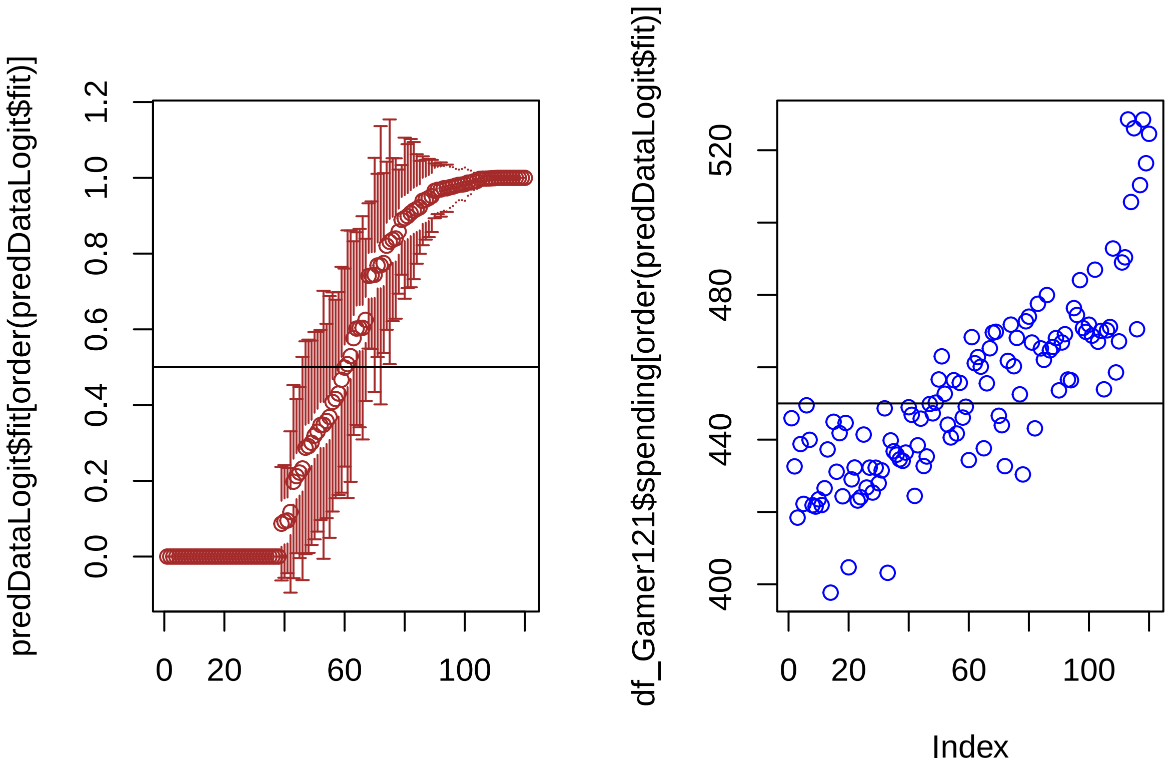
Comment: - The variables in the model have statistically significant impacts on predicting the binary spending are income and age. Because we only have several few significant levels of genres, we cannot conclude that there is significant impact of different genres in predicting the binary spending here. - Income is associated with higher log odds of large spending (gamer's success). This is in line with question 2, where the higher the income is, the higher the monthly spending on game of player1 is, keeping others unchanged. - The effects of age here is positive. The older the player is, the higher the log odds that she/he spends more than 450 per month. This is only in line with player500, but different from the player1 in question 3. - Among the genres of games played, misc leads to higher log odds of large spending than simulation, followed by puzzle. Compared to player1 in question2, the impacts of these 3 genres are similar, playing misc is associated with higher spending than simulation, then puzzle. - Interestingly, the monthly spending in December does not lead to higher log odds of large spending compared to other months, keeping others constant. This is not in line with the results in question 2, where December is positively associated with spending. This may result from the choice of threshold at 450, which is not yet a good cut off for player121 to differentiate between high spending in December versus other months.

5C

```
# Build the prediction probabilities
predDataLogit <- predict(modelLogit, df_Gamer121, type = "response", se.fit = TRUE)
upr <- predDataLogit$fit + predDataLogit$se.fit*1.96 # upper bounds
lwr <- predDataLogit$fit - predDataLogit$se.fit*1.96 # lower bounds
fit <- predDataLogit$fit

# We plot the probabilities in an ascending order so that it is easier for comparison later
plotCI(predDataLogit$fit[order(predDataLogit$fit)],
       ui = upr[order(predDataLogit$fit)], li = lwr[order(predDataLogit$fit)])
```




```r
# To estimate the accuracy of predicted probabilities
# We should see 2 plots side by side: probabilities and real spending of gamer 121

par(mfrow=c(1,2))

plotCI(predDataLogit$fit[order(predDataLogit$fit)], ui = upr[order(predDataLogit$fit)],
       li = lwr[order(predDataLogit$fit)], col = "brown")
abline(h=0.5)

plot(df_Gamer121$spending[order(predDataLogit$fit)], col = "blue")
abline(h = 450)
```



Comment: As can be seen in the graphs, the probabilities estimated are mostly accurate. The higher the probabilities are corresponding to the higher spendings.

5D

- Although the logit model gives us the log odds which are nice to obtain, the choice of threshold can make the model biased to an extent.
- Compare to the results in question 4, here we also obtain the positive effects of income and positive effects of age on spending. However, we do not see the impacts of genre and dummydecember. This can result from the fact that the dependent variable is in binary form with a threshold of spending at 450.
- For example, dummydecember increases spending, but it is not necessarily more likely to upscale the spending from below threshold to pass the threshold compared to other months.
- The same logic can be applied to genre. Compared to question 2, where we also apply modelling for one player, but in that case, the dependent variable is continuous, so the effect can be more easily detected than this binary outcome.
- Also, the lagged variables are not included in this model. Including lagged spending can increase the accuracy and fitness of the model.