```
# whole tidyverse package
library(tidyverse)
# Useful for importing data
library(readr)
#Useful for data wrangling
library(dplyr)
#Useful for data wrangling
library(tidyr)
# Useful for creating tidy tables
library(knitr)
# useful for working with vectors and functions
library(purrr)
# useful to create insightful summaries of data set
library(skimr)
# useful to create insightful reports on data set
library(DataExplorer)
library(ggplot2)

## Importing the data set
games_sales <- read.csv("/Users/rebeccaswedberg/Downloads/game_data/
games_sales.csv", header = TRUE)

# Creating a subset of data for North American Sales
#North American Sales
nadata <- data.frame(games_sales$Year, games_sales$NA_Sales)
names(nadata)[2] <- 'NA_Sales'
names(nadata)[1] <- 'Year'
head(nadata)

summary(nadata)

# Convert Year to integer

nadata$Year <- as.integer(nadata$Year)
summary(nadata)

# sum of missing values, checking the data to see if any missing values
sum(is.na (nadata))

# replace NA with 0
nadata[is.na(nadata)] = 0
head(nadata)

#Removing Years =0
nadata3 = nadata[rowSums(nadata['Year'])>0,]


nadata3 <- aggregate(nadata3["NA_Sales"],by=nadata3["Year"],sum)

head(nadata3)


# Plot the relationship with base R graphics

plot(nadata3$Year, nadata3$NA_Sales)

## Fit the simple linear regression model
modelnadata <-lm(NA_Sales ~ Year, data = nadata3)
```

```
## View the model

modelnadata

## View more outputs for the model - the full regression table
summary(modelnadata)

## Year explains 27% of the variability

## View residuals on a plot

plot(modelnadata$residuals)

## Add line-of-best-fit

abline(coefficients(modelnadata))

## Complete a log transformation with dplyr's mutate() function

nadata3 <- mutate(nadata3,
              logIndex = log(NA_Sales))

## View new object with new variable

head(nadata3)

## Plot the relationship between year and logIndex

plot(nadata3$Year, nadata3$logIndex)

## Make a forecast with this model
## View the last six rows of the data set

tail(nadata3)

## Create a new data frame for the forecast values

nadataForecast <- data.frame(Year = 2021:2022)

## Predict from 2021 to 2022

predict(modelnadata, newdata = nadataForecast)

## Add the values to the cpiForecast data frame

nadataForecast$logIndex <- predict(modelnadata, newdata = nadataForecast)

## Add the actual index as opposed to the log index by exponentiation

nadataForecast <- mutate(nadataForecast,
                      Index = exp(logIndex))

## View the cpiForecast data frame

nadataForecast

ggplot(nadata3,
       mapping = aes(x = Year, y = NA_Sales)) +
```

```
geom_point()
```