

```

# 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 European Sales

eudata <- data.frame(games_sales$Year, games_sales$EU_Sales)
names(eudata)[2] <- 'EU_Sales'
names(eudata)[1] <- 'Year'
head(eudata)

summary(eudata)

# Convert Year to integer

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

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

#Removing Years =0
eudata3 = eudata[rowSums(eudata['Year'])>0,]

sum(is.na (eudata))

eudata3 <- aggregate(eudata3["EU_Sales"],by=eudata3["Year"],sum)

head(eudata3)

# Plot the relationship with base R graphics

plot(eudata3$Year, eudata3$EU_Sales)

## Fit the simple linear regression model
modeleudata <-lm(EU_Sales ~ Year, data = eudata3)

```

```

## View the model

modeleudata

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

## Year explains 41% of the variability

## View residuals on a plot

plot(modeleudata$residuals)

## Add line-of-best-fit

abline(coefficients(modeleudata))

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

eudata3 <- mutate(eudata3,
                  logIndex = log(EU_Sales))

## View new object with new variable

head(eudata3)

## Plot the relationship between year and logIndex

plot(eudata3$Year, eudata3$logIndex)

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

tail(eudata3)

## Create a new data frame for the forecast values

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

## Predict from 2021 to 2022

predict(modeleudata, newdata = eudataForecast)

## Add the values to the cpiForecast data frame

eudataForecast$logIndex <- predict(modeleudata, newdata = eudataForecast)

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

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

## View the cpiForecast data frame

eudataForecast

ggplot(eudata3,

```

```
    mapping = aes(x = Year, y = EU_Sales)) +  
geom_point()
```