ASSIGNMENT COVER SHEET

SUBJECT NUMBER & NAME	36103 Statistical Thinking for Data Science
STUDENT NAMES & IDs (SURNAME, FIRST NAME, STUDENT ID)	Arunachalam, Abishek (SID: 13001262) Jiang, Benjamin (SID: 12875314) Pelayre, Anne Gorge (SID: 13102191) Raghavan, Saminathan (SID:13075597) Ramal, Miguel (SID: 00060259)
TEAM NAME	SKEPTICS
STUDENT EMAIL	abishek.arunachalam@student.uts.edu.au, benjamin.jiang@student.uts.edu.au, annegorge.pelayre@student.uts.edu.au, saminathan.raghavan@student.uts.edu.au, miguel.ramal@student.uts.edu.au
DUE DATE	30 April 2018
ASSESSMENT ITEM NUMBER/TITLE	AT2 Data analysis project, Part A: Project Plan

- I/We confirm that the work submitted conforms with the university's guidelines on academic integrity. *Refer to the UTS policy on 'Advice to Students on Good Academic Practice'*: http://www.gsu.uts.edu.au/policies/academicpractice.html
- I/We am aware of the penalties for plagiarism. This assignment is my/our own work and I/we have not handed in this assignment (either part or completely) for assessment in another subject.
- If this assignment is submitted after the due date I/we understand that it will incur a penalty for lateness unless I have previously had an extension of time approved and have attached the written confirmation of this extension.

Please provide details of extensions granted here if applicable

_____

_____

Signature of Team: _____Skeptics_____ **Date:** 30 / 04 / 2018

If submitted electronically tick here to indicate you agree with the above [X]

Rationale for project

Gross Domestic Product (GDP) represents the total dollar value of all goods and services a country produced over a specific time period, often referred to as the size of the country's economy. As one of the most widely used economic indicators, GDP is used to gauge the health of a country's economy (Investopedia 2018). Given the importance of having a healthy economy to the wellbeing of a country's citizens, our team considered whether it was possible to predict future GDP of Australia using other historical economic and non-economic information (further discussed in datasets section below).

Our team viewed choosing this topic as a learning opportunity, to better understand how we as individuals and as a community can contribute to the economy. As a team of data analysts without any formal qualifications in economics, will try to decode the economic jargon and provide insights on the important factors that influence a country's economy.

Research questions

As there are two types of GDP that economists use to measure a country's economy, our regression model will disregard real GDP (economic output adjusted for the effects of inflation) and solely focus on predicting nominal GDP (a country's economic output without an inflation adjustment). The research questions that we try to answer with the data:

- **Can GDP be accurately predicted given the historical economic and non-economic factors?**
- **Which Economic and Non-Economic factors are most influential to nominal GDP?**
- **Does unemployment rate have an effect on GDP?**

Range of datasets examined and chosen for analysis

After researching information for economic indicators in Australia, most sources including an e-brief article on the Parliament of Australia website (Woods n.d.) indicate the Australian Bureau of Statistics (ABS) as the main source of economic statistics in Australia.

The ABS site provides a free tool: ABS.Stat that offers web browsing and web services interfaces to display and extract data on multiple themes such as Economy, Health, Industry, Labour, People, Census and other snapshots of Australia.

Whilst measuring GDP can be complicated, the calculation can be done in one of three ways: either by adding up what everyone earned in a year (income approach) or by adding up what everyone spent (expenditure method), or by how much everyone produced (production approach). While each approach should conceptually deliver the same estimate of GDP; if the three measures are compiled independently using different data sources, then different estimates of GDP will result (ABS 2012). To combat this issue, the estimates in the GDP data sets had been pre-balanced by the ABS between the three approaches.

As it did not matter which method we choose as long as we were consistent in our logic, we choose the expenditure method. It had the most readily available information for calculating GDP based on the formula:

GDP = Consumption + Investment + Government spending + Net Exports.

In doing so, when choosing our data sets, we were also careful not to choose datasets that were components of each of the methods but rather indicators for the components. For example, we used Consumer Price Index and Business Sales as an indicator of the level of Consumption in the economy.

Expenditure Approach	**Indicator**	**Link**
GDP	GDP	http://stats.oecd.org/restsdmx/sdmx.ashx/GetData/QNA/AUS.B1_GE.CPCARSA.Q/all?startTime=1960-Q1&endTime=2018-Q1
Consumption	Consumer Price Index	http://www.abs.gov.au/ausstats/abs@.nsf/mf/6401.0
	Sales	http://stat.data.abs.gov.au/#
Investment	3-month Monthly Average Interest Rates(%)	https://www.rba.gov.au/statistics/historical-data.html#interest-rates
	Expenditure	http://stat.data.abs.gov.au/#
	Labour Force	http://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/6202.0Mar%202018?OpenDocument
Government spending	Human Development Index(HDI)	http://hdr.undp.org/en/data#
	Unemployment	https://data.oecd.org/unemp/unemployment-rate.htm
Net Exports	Balance on Goods and Services	http://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/5368.0Feb%202018?OpenDocument
	Exchange rates	https://www.rba.gov.au/statistics/historical-data.html#exchange-rates

Data Sources

- **Gross Domestic Product & Unemployment** – data sourced from The Organisation for Economic Co-operation and Development (OECD).
- **Human Development Index** – data sourced from the United Nations Development Programme (UNDP).
- **Interest Rates & Exchange Rates** – data sourced from the Reserve Bank of Australia (RBA).

- **CPI, Sales, Expenditure, Labour Force & Balance on Goods and Services**- data sourced from Australian Bureau of Statistics (ABS).

Regression modelling techniques

The data being analysed for this project is made up of historical records of numeric values over a number of years, and multiple predictors to consider for forecasting the GDP. Time series models would be the obvious choice as the data vary with time.

Although it is early to pin-point a specific modelling technique, we plan to start with simple models like multiple linear regression and make predictions on test dataset to see how they perform. We plan to pick the important predictors using Lasso regression or use the Principal Components from Principal Component Analysis (PCA). We are also planning to explore time-series forecasting techniques such as: Auto-Regression (AR) models, Simple Moving Average (SMA), Exponential Smoothing (SES), Autoregressive Integration Moving Average (ARIMA), Recurrent Neural Network (RNN) and Holt-winters.

Issues that may arise during project

There are some NA values in the dataset. Decision has to be made whether to remove them or to calculate justified aggregates and use them for analysis. Each predictor is on a different scale, so they need to be standardised before they are used for Principal Component Analysis. There is a chance that some of the predictors do not show any correlation to GDP. In that case the poor predictors need to be dropped and modelling needs to be performed with the most influential predictors. If there are very few influential predictors, then new variables need to be added in their place.

Forecasting a model for GDP implies making decision on the time horizon of predictions. A shorter time horizon would be easier to predict with higher confidence. This also leads into another aspect of the forecast on how frequent could the forecast be updated over time as new information becomes available (assuming latest information would imply more accurate predictions).

References

Australian Bureau of Statistics 2012, *Defining and measuring GDP*, viewed 28 April 2018, <http://www.abs.gov.au/ausstats/abs@.nsf/Lookup/by%20Subject/1301.0~2012~Main%20Features~Defining%20and%20measuring%20GDP~221>.

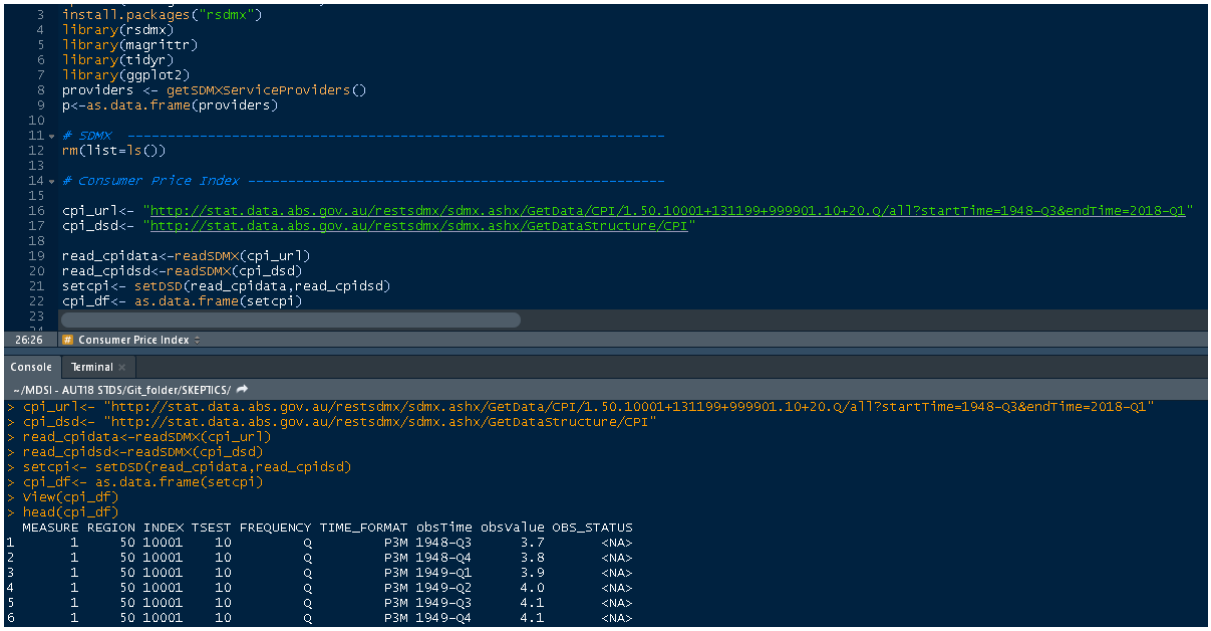
Investopedia 2018, *What is GDP and why is it so important to economists and investors?*, viewed 28 April 2018, < https://www.investopedia.com/ask/answers/199.asp>.

Woods, G. n.d., 'Economic Indicators on the Internet', *Economic Indicators on the Internet*, E-Brief, viewed 26 April 2018, <https://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/Publications_Archive/archive/ecindicators>.

Appendix

Below is a snippet from a script using API methods for data extraction from the Australian Bureau of Statistics. SDMX API was used to import some of the datasets in R.

Obtaining Consumer Price Index data:

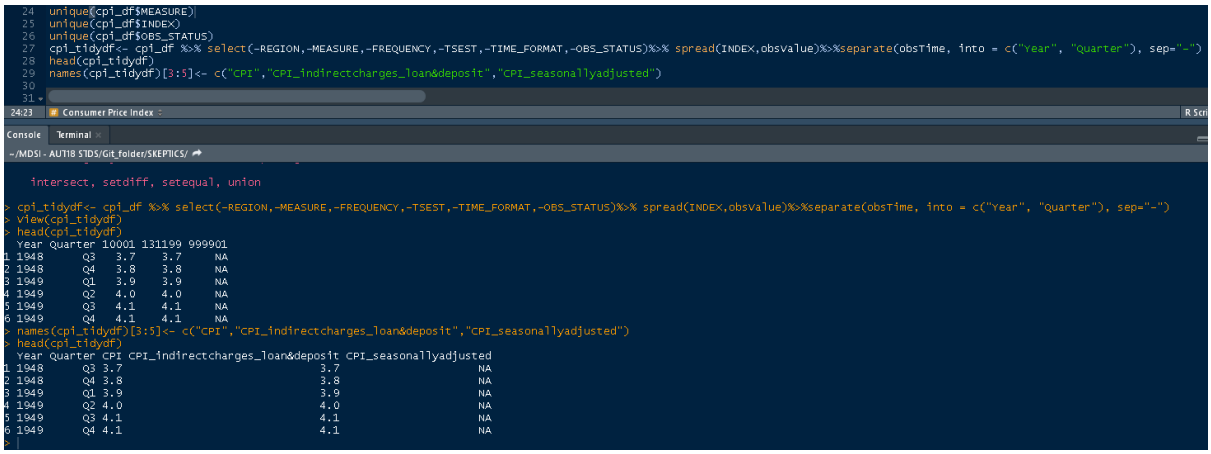


```
install.packages("rsdmx")
library(rsdmx)
library(magrittr)
library(tidyr)
library(ggplot2)
providers <- getSDMXServiceProviders()
p<-as.data.frame(providers)

# SDMX  ----------------------------------------------------------------------
rm(list=ls())

# Consumer Price Index -------------------------------------------------------

cpi_url<- "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetData/CPI/1.50.10001+131199+999901.10+20.Q/all?startTime=1948-Q3&endTime=2018-Q1"
cpi_dsd<- "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetDataStructure/CPI"

read_cpidata<-readSDMX(cpi_url)
read_cpidsd<-readSDMX(cpi_dsd)
setcpi<- setDSD(read_cpidata,read_cpidsd)
cpi_df<- as.data.frame(setcpi)
```

```
> cpi_url<- "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetData/CPI/1.50.10001+131199+999901.10+20.Q/all?startTime=1948-Q3&endTime=2018-Q1"
> cpi_dsd<- "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetDataStructure/CPI"
> read_cpidata<-readSDMX(cpi_url)
> read_cpidsd<-readSDMX(cpi_dsd)
> setcpi<- setDSD(read_cpidata,read_cpidsd)
> cpi_df<- as.data.frame(setcpi)
> view(cpi_df)
> head(cpi_df)
  MEASURE REGION INDEX TSEST FREQUENCY TIME_FORMAT obsTime obsValue OBS_STATUS
1       1     50 10001    10         Q          P3M 1948-Q3      3.7       <NA>
2       1     50 10001    10         Q          P3M 1948-Q4      3.8       <NA>
3       1     50 10001    10         Q          P3M 1949-Q1      3.9       <NA>
4       1     50 10001    10         Q          P3M 1949-Q2      4.0       <NA>
5       1     50 10001    10         Q          P3M 1949-Q3      4.1       <NA>
6       1     50 10001    10         Q          P3M 1949-Q4      4.1       <NA>
```

As data does not come in the format required to merge or analyse properly (see data format on screenshot above), we are required to transform obtained datasets to prepare data-frames ready for analysis and data merging.

Transformation of data (consumer price index) into usable data frame for project use



```
unique(cpi_df$MEASURE)
unique(cpi_df$INDEX)
unique(cpi_df$OBS_STATUS)
cpi_tidydf<- cpi_df %>% select(-REGION,-MEASURE,-FREQUENCY,-TSEST,-TIME_FORMAT,-OBS_STATUS)%>% spread(INDEX,obsValue)%>%separate(obsTime, into = c("Year", "Quarter"), sep="-")
head(cpi_tidydf)
names(cpi_tidydf)[3:5]<- c("CPI","CPI_indirectcharges_loan&deposit","CPI_seasonallyadjusted")
```

```
    intersect, setdiff, setequal, union

> cpi_tidydf<- cpi_df %>% select(-REGION,-MEASURE,-FREQUENCY,-TSEST,-TIME_FORMAT,-OBS_STATUS)%>% spread(INDEX,obsValue)%>%separate(obsTime, into = c("Year", "Quarter"), sep="-")
> view(cpi_tidydf)
> head(cpi_tidydf)
  Year Quarter 10001 131199 999901
1 1948      Q3   3.7    3.7     NA
2 1948      Q4   3.8    3.8     NA
3 1949      Q1   3.9    3.9     NA
4 1949      Q2   4.0    4.0     NA
5 1949      Q3   4.1    4.1     NA
6 1949      Q4   4.1    4.1     NA
> names(cpi_tidydf)[3:5]<- c("CPI","CPI_indirectcharges_loan&deposit","CPI_seasonallyadjusted")
> head(cpi_tidydf)
  Year Quarter CPI CPI_indirectcharges_loan&deposit CPI_seasonallyadjusted
1 1948      Q3 3.7                              3.7                     NA
2 1948      Q4 3.8                              3.8                     NA
3 1949      Q1 3.9                              3.9                     NA
4 1949      Q2 4.0                              4.0                     NA
5 1949      Q3 4.1                              4.1                     NA
6 1949      Q4 4.1                              4.1                     NA
```

In similar fashion, other datasets have been obtained and transformed using same methods:

```
# House Price Index ------------------------------------------------------

hpi_url<- "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetData/RES_PROP_INDEX/1.3+2+1.100.Q/all?startTime=2002-Q1&

hpi_dsd<- "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetDataStructure/RES_PROP_INDEX"

read_hpidata<- readSDMX(hpi_url)
read_hpidsd<- readSDMX(hpi_dsd)
sethpi<- setDSD(read_hpidata,read_hpidsd)
hpi_df<- as.data.frame(sethpi)
hpi_tidydf<-hpi_df %<>% select(-MEASURE,-ASGS_2011,-FREQUENCY,-TIME_FORMAT)
head(hpi_tidydf)
hpi_tidydf<- separate(hpi_tidydf, obsTime, into = c("Year", "Quarter"), sep="-")
hpi_tidydf<- spread(hpi_tidydf,PROP_TYPE,obsValue)
names(hpi_tidydf)[3:5]<- c("HPI_residentprop","HPI_establishedhouse","HPI_dwelling")

# Export Price Index ANZSIC ----------------------------------------------

epi_url<- "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetData/ITPI_EXPORT/1.8123922+8123923+8123924.Q/all?startTi

epi_dsd<- "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetDataStructure/ITPI_EXPORT"

read_epidata<- readSDMX(epi_url)
read_epidsd<- readSDMX(epi_dsd)
setepi<-setDSD(read_epidata,read_epidsd)
epi_df<-as.data.frame(setepi)
head(epi_df)
unique(epi_df$TIME_FORMAT)
epi_tidydf<- epi_df %>% select(-MEASURE,-FREQUENCY,-TIME_FORMAT) %>% spread(INDEX,obsValue) %>% separate(obsTime, int
head(epi_tidydf)
names(epi_tidydf)[3:5]<- c("EXPI_ANZSIC_agri_forest_fishing","EXPI_ANZSIC_Mining","EXPI_ANZSIC_Manufacturing")
```

As GDP is derived from multiple economic indicators, and each is composed of multiple indices (derived from multiple datasets), we are employing data-merging techniques using left joins on year and quarter to match corresponding indices over a time-series spread of data.

Merging multiple data-frames on consumer price index into a master data-frame for project

```
# Final DataFrame ------------------------------------------------------

getwd()
final_df<-cpi_tidydf %>% left_join(hpi_tidydf, by = c("Year" = "Year", "Quarter" = "Quarter")) %>% left_join(epi_tidydf, by = c("Year" = "Year", "Quarter" = "Quarter")) %>%
  left_join(expi_tidydf, by = c("Year" = "Year", "Quarter" = "Quarter")) %>% left_join (imi_tidydf, by = c("Year" = "Year", "Quarter" = "Quarter")) %>%
  left_join(impi_tidydf,by = c("Year" = "Year", "Quarter" = "Quarter")) %>% left_join (wpi_tidydf, by = c("Year" = "Year", "Quarter" = "Quarter")) %>%
  left_join(ppi_tidydf, by = c("Year" = "Year", "Quarter" = "Quarter"))
head(final_df)
write.csv(file="priceindex.csv",final_df)
```

```
150:1  Final DataFrame :                                                    R Scrip

Console  Terminal
~/MDSI - AUT18 STDS/Git_folder/SKEPTICS/
> tail(final_df)
    Year Quarter   CPI CPI_indirectcharges_loan&deposit CPI_seasonallyadjusted HPI_residentprop HPI_establishedhouse HPI_dwelling EXPI_ANZSIC_agri_forest_fishing
274 2016      Q4 110.0                             110.5                   109.8            131.3               143.8         140.6                          96.5
275 2017      Q1 110.5                             111.1                   110.5            133.5               147.3         143.7                          97.4
276 2017      Q2 110.7                             111.4                   110.9            136.1               150.1         146.5                          97.2
277 2017      Q3 111.4                             112.2                   111.3            135.0               150.1         146.2                          98.3
278 2017      Q4 112.1                             113.1                   112.0            135.8               151.8         147.6                         102.7
279 2018      Q1 112.6                             113.6                   112.6               NA                  NA            NA                         106.8
    EXPI_ANZSIC_Mining EXPI_ANZSIC_Manufacturing EXPI_SITC IMPI_SITC IMPI_ANZSIC_agri_forest_fishing IMPI_ANZSIC_Mining IMPI_ANZSIC_Manufacturing   WPI WPI_seasonal WPI_trend
274               76.1                     106.1      88.2     102.7                           126.4              57.4                     107.2 125.1       125.0     125.0
275               86.3                     109.6      96.0     103.9                           132.3              63.6                     107.9 125.6       125.7     125.6
276               77.5                     110.2      90.5     103.8                           131.5              59.6                     108.2 126.1       126.3     126.3
277               74.5                     107.6      87.8     102.1                           129.1              58.9                     106.3 127.1       126.9     126.9
278               75.7                     111.9      90.3     104.1                           132.1              65.9                     107.9 127.7       127.6     127.6
279               81.9                     113.0      94.7     106.3                           133.4              74.9                     109.6    NA          NA        NA
    PPI_incl_exports PPI_excl_exports
274            104.4            106.8
275            105.9            107.3
276            105.8            107.8
277            105.5            108.0
278            106.8            108.6
279               NA               NA
>
```

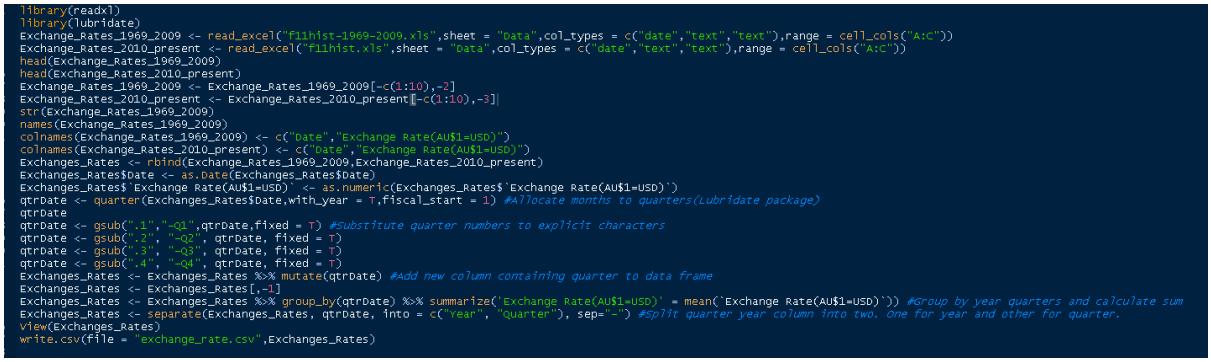
Another sample of data-transformation can be observed on script below where dataset obtained on balance of goods and services was recorded on a monthly frequency and we are required to transform the values into quartile equivalents in order to merge with other datasets

Transforming a monthly frequency index into quarters

```
library(readxl)
library(dplyr)
library(lubridate)
Balance_on_goods_and_services <- read_excel("Balance on goods and services (5368m) Column B.xls",
                                            sheet = "Data1", col_types = c("date",
                                                                           "text", "text", "text", "text", "text",
                                                                           "text", "text", "text", "text", "text",
                                                                           "text", "text", "text", "text", "text"))

# create data-frame with desired rows and columns from original dataset
mydata <- Balance_on_goods_and_services[-c(1:9),-c(3:16)]
str(mydata) # Balance is in char format and date is in POSIXct format
names(mydata)
# Rename columns
colnames(mydata) <- c("Date","Balance_on_goods_and_services")

mydata$Date <- as.Date(mydata$Date) #Convert to date format
mydata$Balance_on_goods_and_services <- as.numeric(mydata$Balance_on_goods_and_services) # Convert Balance to numeric

qtrDate <- quarter(mydata$Date,with_year = T,fiscal_start = 1) #Allocate months to quarters(Lubridate package)
qtrDate <- gsub(".1", "-Q1",qtrDate,fixed = T) #Substitute quarter numbers to explicit characters
qtrDate <- gsub(".2", "-Q2", qtrDate, fixed = T)
qtrDate <- gsub(".3", "-Q3", qtrDate, fixed = T)
qtrDate <- gsub(".4", "-Q4", qtrDate, fixed = T)
mydata <- mydata %>% mutate(qtrDate) #Add new column containing quarter to data frame
mydata <- mydata[,-1] #Remove date column
mydata <- mydata %>% group_by(qtrDate) %>% summarize(Balance_on_goods_and_services = sum(Balance_on_goods_and_services)) #Group by year quarters and calculate sum
Balance_on_goods_and_services <- separate(mydata, qtrDate, into = c("Year", "Quarter"), sep="-") #Split quarter year column into two. One for year and other for quarter.
colnames(Balance_on_goods_and_services)[3] <- "Balance_on_goods_and_services($ Million)"
View(Balance_on_goods_and_services)
write.csv(file = "balance_on_goods_and_services.csv",Balance_on_goods_and_services)
```

Some of the data required for analysis does not come from data sources offering API interfaces to obtain them therefore we are required to download and import the csv, xls dataset and manipulate as required.

Importing downloaded (.xls) dataset and transformation to quarters



```
library(readxl)
library(lubridate)
Exchange_Rates_1969_2009 <- read_excel("f11hist-1969-2009.xls",sheet = "Data",col_types = c("date","text","text"),range = cell_cols("A:C"))
Exchange_Rates_2010_present <- read_excel("f11hist.xls",sheet = "Data",col_types = c("date","text","text"),range = cell_cols("A:C"))
head(Exchange_Rates_1969_2009)
head(Exchange_Rates_2010_present)
Exchange_Rates_1969_2009 <- Exchange_Rates_1969_2009[-c(1:10),-2]
Exchange_Rates_2010_present <- Exchange_Rates_2010_present[-c(1:10),-3]
str(Exchange_Rates_1969_2009)
names(Exchange_Rates_1969_2009)
colnames(Exchange_Rates_1969_2009) <- c("Date","Exchange Rate(AU$1=USD)")
colnames(Exchange_Rates_2010_present) <- c("Date","Exchange Rate(AU$1=USD)")
Exchanges_Rates <- rbind(Exchange_Rates_1969_2009,Exchange_Rates_2010_present)
Exchanges_Rates$Date <- as.Date(Exchanges_Rates$Date)
Exchanges_Rates$`Exchange Rate(AU$1=USD)` <- as.numeric(Exchanges_Rates$`Exchange Rate(AU$1=USD)`)
qtrDate <- quarter(Exchanges_Rates$Date,with_year = T,fiscal_start = 1) #Allocate months to quarters(Lubridate package)
qtrDate
qtrDate <- gsub(".1","-Q1",qtrDate,fixed = T) #Substitute quarter numbers to explicit characters
qtrDate <- gsub(".2", "-Q2", qtrDate, fixed = T)
qtrDate <- gsub(".3", "-Q3", qtrDate, fixed = T)
qtrDate <- gsub(".4", "-Q4", qtrDate, fixed = T)
Exchanges_Rates <- Exchanges_Rates %>% mutate(qtrDate) #Add new column containing quarter to data frame
Exchanges_Rates <- Exchanges_Rates[,-1]
Exchanges_Rates <- Exchanges_Rates %>% group_by(qtrDate) %>% summarize('Exchange Rate(AU$1=USD)' = mean(`Exchange Rate(AU$1=USD)`)) #Group by year quarters and calculate sum
Exchanges_Rates <- separate(Exchanges_Rates, qtrDate, into = c("Year", "Quarter"), sep="-") #Split quarter year column into two. One for year and other for quarter.
View(Exchanges_Rates)
write.csv(file = "exchange_rate.csv",Exchanges_Rates)
```

Code:

```
# First Steps loading Required Packages ----------------------------------


#rm(list=ls())
check.packages <- function(pkg){
    new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
    if (length(new.pkg))
     install.packages(new.pkg, dependencies = TRUE)
    sapply(pkg, require, character.only = TRUE)
}
```

```
packages<-c("ggplot2", "rsdmx", "magrittr", "tidyr", "readxl",
"dplyr","lubridate") #required packages
check.packages(packages) # Calling check.packages function


options(stringsAsFactors = FALSE)


# SDMX ----------------------------------------------------------------------


providers <- getSDMXServiceProviders()
p<-as.data.frame(providers)


getwd()


# Balance of goods and Services in Millions ------------------------------


Balance_on_goods_and_services <- read_excel("536802.xls",
                                            sheet = "Data1", col_types =
c("date",

"text"),range = cell_cols("A:B"))


# create data-frame with desired rows and columns from original dataset
mydata <- Balance_on_goods_and_services[-c(1:9),]
str(mydata) # Balance is in char format and date is in POSIXct format
names(mydata)
# Rename columns
colnames(mydata) <- c("Date","Balance_on_goods_and_services")


mydata$Date <- as.Date(mydata$Date) #Convert to date format
mydata$Balance_on_goods_and_services <-
as.numeric(mydata$Balance_on_goods_and_services) # Convert Balance to numeric


qtrDate <- quarter(mydata$Date,with_year = T,fiscal_start = 1) #Allocate months
to quarters(Lubridate package)
qtrDate <- gsub(".1","-Q1",qtrDate,fixed = T) #Substitute quarter numbers to
explicit characters
qtrDate <- gsub(".2", "-Q2", qtrDate, fixed = T)
```

```
qtrDate <- gsub(".3", "-Q3", qtrDate, fixed = T)
qtrDate <- gsub(".4", "-Q4", qtrDate, fixed = T)
mydata <- mydata %>% mutate(qtrDate) #Add new column containing quarter to data
frame
mydata <- mydata[,-1] #Remove date column
mydata <- mydata %>% group_by(qtrDate) %>%
summarize(Balance_on_goods_and_services = sum(Balance_on_goods_and_services))
#Group by year quarters and calculate sum
Balance_on_goods_and_services <- separate(mydata, qtrDate, into = c("Year",
"Quarter"), sep="-") #Split quarter year column into two. One for year and other
for quarter.
colnames(Balance_on_goods_and_services)[3] <- "Balance_on_goods_and_services($
Million)"
head(Balance_on_goods_and_services)
glimpse(Balance_on_goods_and_services)
#write.csv(file =
"balance_on_goods_and_services.csv",Balance_on_goods_and_services)



# Business Indicators ----------------------------------------------------


url <-
"http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetData/QBIS/10+50+90+110.TOTAL.0
.99.10+20+30.Q/all?startTime=1985-Q1&endTime=2017-Q4"
business_indicators <- readSDMX(url,dsd = T)
business_indicators <- as.data.frame(business_indicators)
business_indicators <-
business_indicators[,c("MEASURE","TSEST","obsTime","obsValue")]
head(business_indicators)


unique(business_indicators$MEASURE)
# Measure - 10-Sales($ Million), 50-Inventories($ Million), 90-Wages, 110 - Gross
Operating Profits
unique(business_indicators$TSEST)
#TSEST - 10-Original, 20-Seasonally Adjusted, 30-Trend
business_indicators$MEASURE[business_indicators$MEASURE == '10'] <- "Sales($
Million)"
business_indicators$MEASURE[business_indicators$MEASURE == '50'] <-
"Inventories($ Million)"
business_indicators$MEASURE[business_indicators$MEASURE == '90'] <- "Wages($
Million)"
```

```
business_indicators$MEASURE[business_indicators$MEASURE == '110'] <- "Gross
Operating Profit($ Million)"
unique(business_indicators$MEASURE)
#business_indicators %>% count(MEASURE) #Inventories seem to be inconsistent with
others
#business_indicators$MEASURE <- as.factor(business_indicators$MEASURE)
business_indicators$TSEST[business_indicators$TSEST== '10'] <- "Original" #132
business_indicators$TSEST[business_indicators$TSEST== '20'] <- "Seasonally
adjusted" #68
business_indicators$TSEST[business_indicators$TSEST == '30'] <- "Trend" #68
business_indicators_tidy_df<- business_indicators %>%
dplyr::filter(TSEST=="Original") %>%
    spread(MEASURE,obsValue) %>%
    separate( obsTime, into = c("Year", "Quarter"), sep="-") %>%
    select(-TSEST)
head(business_indicators_tidy_df)


#Expenditure----------------------------------------------------


url <- "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetData/CAPEX/1.999.-
.0.10+20+30.Q/all?startTime=1987-Q2&endTime=2017-Q4"
Expenditure <- as.data.frame(readSDMX(url))
head(Expenditure)
unique(Expenditure$STATE)
Expenditure$TSEST[Expenditure$TSEST == '10'] <- "Original" #132
Expenditure$TSEST[Expenditure$TSEST == '20'] <- "Seasonally adjusted" #68
Expenditure$TSEST[Expenditure$TSEST == '30'] <- "Trend" #68


# Expenditure.Original <- filter(GOP, TSEST == "Original")$obsValue
# Expenditure.SeasonallyAdjusted <- filter(GOP, TSEST == "SeasonallyAdjusted")
# Expenditure.Trend <- filter(GOP, TSEST == "Trend")
head(Expenditure)
Expenditure_tidydf<- Expenditure %>% dplyr::filter(TSEST=="Original") %>%
    separate( obsTime, into = c("Year", "Quarter"), sep="-") %>%
    select(-TSEST,-EXP,-STATE,-IND,-FREQUENCY,-TIME_FORMAT,-OBS_STATUS,-ASSET)
colnames(Expenditure_tidydf)[3] <- "Expenditure($ Million)"
head(Expenditure)


# Exchange Rates ------------------------------------------------------------
```

```
Exchange_Rates_1969_2009 <- readxl::read_excel("f11hist-1969-2009.xls",sheet =
"Data",col_types = c("date","text","text"),range = cell_cols("A:C"))
Exchange_Rates_2010_present <- read_excel("f11hist.xls",sheet = "Data",col_types
= c("date","text","text"),range = cell_cols("A:C"))
head(Exchange_Rates_1969_2009)
head(Exchange_Rates_2010_present)
Exchange_Rates_1969_2009 <- Exchange_Rates_1969_2009[-c(1:10),-2]
Exchange_Rates_2010_present <- Exchange_Rates_2010_present[-c(1:10),-3]
str(Exchange_Rates_1969_2009)
names(Exchange_Rates_1969_2009)
colnames(Exchange_Rates_1969_2009) <- c("Date","Exchange Rate(AU$1=USD)")
colnames(Exchange_Rates_2010_present) <- c("Date","Exchange Rate(AU$1=USD)")
Exchanges_Rates <- rbind(Exchange_Rates_1969_2009,Exchange_Rates_2010_present) #
Adding Rows from both the sheets
Exchanges_Rates$Date <- as.Date(Exchanges_Rates$Date)
Exchanges_Rates$`Exchange Rate(AU$1=USD)` <- as.numeric(Exchanges_Rates$`Exchange
Rate(AU$1=USD)`)
qtrDate <- quarter(Exchanges_Rates$Date,with_year = T,fiscal_start = 1) #Allocate
months to quarters(Lubridate package)
qtrDate
qtrDate <- gsub(".1","-Q1",qtrDate,fixed = T) #Substitute quarter numbers to
explicit characters
qtrDate <- gsub(".2", "-Q2", qtrDate, fixed = T)
qtrDate <- gsub(".3", "-Q3", qtrDate, fixed = T)
qtrDate <- gsub(".4", "-Q4", qtrDate, fixed = T)
Exchanges_Rates <- Exchanges_Rates %>% mutate(qtrDate) #Add new column containing
quarter to data frame
Exchanges_Rates <- Exchanges_Rates[,-1]
Exchanges_Rates_tidydf <- Exchanges_Rates %>%
    group_by(qtrDate) %>%
    summarize('Exchange Rate(AU$1=USD)' = mean(`Exchange Rate(AU$1=USD)`)) #Group
by year quarters and calculate mean


Exchanges_Rates_tidydf <- separate(Exchanges_Rates_tidydf, qtrDate, into =
c("Year", "Quarter"), sep="-") #Split quarter year column into two. One for year
and other for quarter.
head(Exchanges_Rates_tidydf)
#write.csv(file = "exchange_rate.csv",Exchanges_Rates)



# GDP ----------------------------------------------------------------------
```

```
url <-
"http://stats.oecd.org/restsdmx/sdmx.ashx/GetData/QNA/AUS.B1_GE.CPCARSA.Q/all?sta
rtTime=1960-Q1&endTime=2018-Q1"
GDP <- as.data.frame(readSDMX(url,dsd = T))
str(GDP)


GDP <- GDP[,c("obsTime","obsValue")]
head(GDP,60)
GDP <- separate(GDP,obsTime, into = c("Year", "Quarter"), sep="-")
colnames(GDP)[3] <- "GDP(US$ Millions)"
View(GDP)


#write.csv(file = "GDP.csv",GDP)



# Human Development Index ------------------------------------------------


HumanDevelopmentIndex <- read.csv("Human Development Index (HDI).csv",header = F)
head(HumanDevelopmentIndex)
HumanDevelopmentIndex <- filter(HumanDevelopmentIndex,V2 == "Country" | V2=="
Australia")
HumanDevelopmentIndex
Years <- 1990:2015
HDI <- as.numeric(HumanDevelopmentIndex[2,-c(1,2)])
HDI <- data.frame(Year = Years,HDI = HDI )
names(HDI)[2] <- "HDI(%)"
HDI <- HDI[rep(seq_len(nrow(HDI)), each=4),]
rownames(HDI) <- c()
Quarter <- rep(c("Q1","Q2","Q3","Q4"),length(Years)) #Create quarters
HDI <- cbind(Quarter,HDI)
HDI <- HDI[,c(2,1,3)] #Reorder columns
summary(HDI)
HDI$Year<-as.character(HDI$Year)
head(HDI)
#write.csv(file="human_development_index.csv",HDI)



# Interest Rates ---------------------------------------------------------
```

```
interestRates <- read_excel("f01hist.xls",sheet = "Data",col_types =
c("date","text","text","text","text","text"),range = cell_cols("A:F"))
interestRates <- interestRates[-c(1:10),c(1,5)]
head(interestRates,n=15)
#3-month BABs/NCDs Bank Accepted Bills/Negotiable Certificates of Deposit-3
months; monthly average
colnames(interestRates) <- c("Date","Interest rates")
interestRates$Date <- as.Date(interestRates$Date)
interestRates$`Interest rates` <- as.numeric(interestRates$`Interest rates`)
qtrDate <- quarter(interestRates$Date,with_year = T,fiscal_start = 1) #Allocate
months to quarters(Lubridate package)
qtrDate
qtrDate <- gsub(".1","-Q1",qtrDate,fixed = T) #Substitute quarter numbers to
explicit characters
qtrDate <- gsub(".2", "-Q2", qtrDate, fixed = T)
qtrDate <- gsub(".3", "-Q3", qtrDate, fixed = T)
qtrDate <- gsub(".4", "-Q4", qtrDate, fixed = T)
interestRates <- interestRates %>% mutate(qtrDate) #Add new column containing
quarter to data frame
interestRates <- interestRates[,-1]
interestRates <- interestRates %>%
    group_by(qtrDate) %>%
    summarize(Interest_Rates= mean(`Interest rates`))


interestRates <- separate(interestRates, qtrDate, into = c("Year", "Quarter"),
sep="-") #Split quarter year column into two. One for year and other for quarter.
head(interestRates)
colnames(interestRates)[3] <- "3-month Monthly Average Interest Rates(%)"
#write.csv(file="interest_rates.csv",interestRates)



# Unemployment Rate ----------------------------------------------------


unemployment <- read.csv("DP_LIVE_28042018200453939.csv",header = T)
names(unemployment)[1]<- "LOCATION"
unemployment.AUS <- filter(unemployment,LOCATION =="AUS")
unemployment.AUS <- unemployment.AUS[,c("TIME","Value")]
unemployment.AUS <- separate(unemployment.AUS, TIME, into = c("Year", "Quarter"),
sep="-")
colnames(unemployment.AUS)[3] <- c("Percentage unemployed %")
head(unemployment.AUS)
```

```
#write.csv(file="unemployment.csv",unemployment.AUS)


# Consumer Price Index ------------------------------------------------


cpi_url<-
"http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetData/CPI/1+2+3.50.10001+999901
.10+20.Q/all?startTime=1948-Q3&endTime=2018-Q1"
cpi_dsd<- "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetDataStructure/CPI"


read_cpidata <-readSDMX(cpi_url)
read_cpidsd <-readSDMX(cpi_dsd)
setcpi <- setDSD(read_cpidata,read_cpidsd)
cpi_df <- as.data.frame(setcpi)
head(cpi_df)
unique(cpi_df$MEASURE) # 1- Index 2-Percentage change within quarters 3-
Percentage change from corresponding quarter of Previous year
unique(cpi_df$TSEST) # 10- Original 20- Seasonal
unique(cpi_df$INDEX) # 10001 All groups CPI 999901 All groups Season data


# Fetching original CPI data only ------------------------------------------


cpi_df<- cpi_df %>% filter(TSEST=="10",INDEX=="10001", MEASURE %in% c("1"))


cpi_tidydf <- cpi_df %>%
    select(-REGION,-FREQUENCY,-TSEST,-TIME_FORMAT,-OBS_STATUS,-INDEX)%>%
    spread(MEASURE,obsValue)%>%
    separate(obsTime, into = c("Year", "Quarter"), sep="-")
head(cpi_tidydf)
names(cpi_tidydf)[3] <- c("Consumer Price Index(CPI)")
head(cpi_tidydf)




# Labourforce ---------------------------------------------------------


Labourforce <- read_excel("6202001.xls",
                                        sheet =
"Data1",skip=9,col_types="date",range=cell_cols("A"))
```

```
Labourforce1 <- read_excel("6202001.xls",
                           sheet = "Data1",col_types="text",range=cell_cols("CP"))
# create data-frame with desired rows and columns from original dataset
Labourforce <- na.omit(Labourforce)
Labourforce1 <- Labourforce1[-c(1:9),]
Labourdata <- cbind(Labourforce,Labourforce1)
View(Labourdata)
names(Labourdata) <- c("Date","Labourforce") # Rename columns
str(Labourdata)
Labourdata$Date <- as.Date(Labourdata$Date) #Convert to date format
Labourdata$Labourforce <- as.numeric(Labourdata$Labourforce) # Convert Balance to
numeric


qtrDate <- quarter(Labourdata$Date,with_year = T,fiscal_start = 1) #Allocate
months to quarters(Lubridate package)
qtrDate <- gsub(".1","-Q1",qtrDate,fixed = T) #Substitute quarter numbers to
explicit characters
qtrDate <- gsub(".2", "-Q2", qtrDate, fixed = T)
qtrDate <- gsub(".3", "-Q3", qtrDate, fixed = T)
qtrDate <- gsub(".4", "-Q4", qtrDate, fixed = T)
Labourdata <- Labourdata %>% mutate(qtrDate) #Add new column containing quarter
to data frame
Labourdata <- Labourdata[,-1] #Remove date column
Labourdata <- Labourdata %>%
    group_by(qtrDate) %>%
    summarize(Labourforce = mean(Labourforce)) #Group by year quarters and
calculate sum
Labourdata <- separate(Labourdata, qtrDate, into = c("Year", "Quarter"), sep="-")
#Split quarter year column into two. One for year and other for quarter.
head(Labourdata)


# Preparing Master Datasheet ---------------------------------------------


join_df <- cpi_tidydf %>%
    left_join(business_indicators_tidy_df,by=c("Year" = "Year", "Quarter" =
"Quarter")) %>%
    left_join(Expenditure_tidydf,by=c("Year" = "Year", "Quarter" = "Quarter"))
%>%
    left_join(unemployment.AUS,by=c("Year" = "Year", "Quarter" = "Quarter"))  %>%
    left_join(interestRates,by=c("Year" = "Year", "Quarter" = "Quarter")) %>%
    left_join(HDI,by=c("Year" = "Year", "Quarter" = "Quarter")) %>%
    left_join(GDP,by=c("Year" = "Year", "Quarter" = "Quarter")) %>%
```

```
    left_join(Balance_on_goods_and_services,by=c("Year" = "Year", "Quarter" =
"Quarter")) %>%
    left_join(Exchanges_Rates_tidydf,by=c("Year" = "Year", "Quarter" =
"Quarter")) %>%
    left_join(Labourdata,by=c("Year" = "Year", "Quarter" = "Quarter"))


head(join_df)


# Filtering data from 1970 ------------------------------------------------


master_df <- join_df %>% filter(Year >= "1970")
write.csv(file="master.csv",master_df)
```