

# GIE Webscrapping

*Carlos Espeleta y Erick Quispe*

*05/06/2017*

# Contents

<b>1</b>	<b>Introducción</b>	<b>3</b>
1.1	Motivación . . . . .	3
1.2	Estructura de los datos . . . . .	3
<b>2</b>	<b>Accediendo a los datos</b>	<b>4</b>
2.1	Extracción de los datos de toda la semana . . . . .	5
2.2	Extracción de los datos de múltiples fechas . . . . .	7
2.3	Unión de todo lo anterior: Ejemplo completo . . . . .	9
<b>3</b>	<b>Análisis de los datos</b>	<b>10</b>

May 2, 2017							Up Next	Filter
Date	1:21pm	Currency	Impact	Detail	Actual	Forecast	Previous	Graph
Tue May 2	12:30am	AUD		Cash Rate	1.50%	1.50%	1.50%	
		AUD		RBA Rate Statement				
	1:00am	JPY		BOJ Core CPI y/y	-0.1%	0.2%	0.1%	
	3:15am	EUR		Spanish Manufacturing PMI	54.5	54.3	53.9	
	3:30am	CHF		Manufacturing PMI	57.4	58.2	58.6	
	3:45am	EUR		Italian Manufacturing PMI	56.2	55.9	55.7	
	3:50am	EUR		French Final Manufacturing PMI	55.1	55.1	55.1	
	3:55am	EUR		German Final Manufacturing PMI	58.2	58.2	58.2	
	4:00am	EUR		Final Manufacturing PMI	56.7	56.8	56.8	
		EUR		Italian Monthly Unemployment Rate	11.7%	11.6%	11.5%	
	4:30am	GBP		Manufacturing PMI	57.3	54.0	54.2	
	5:00am	EUR		Unemployment Rate	9.5%	9.4%	9.5%	
	10:32am	NZD		GDT Price Index	3.6%		3.1%	
	All Day	USD		Total Vehicle Sales	16.9M	17.1M	16.6M	
	6:45pm	NZD		Employment Change q/q	1.2%	0.8%	0.7%	
		NZD		Unemployment Rate	4.9%	5.1%	5.2%	
		NZD		Labor Cost Index q/q	0.4%	0.5%	0.4%	

Figure 1: Calendario, vista general

# 1 Introducción

## 1.1 Motivación

El objetivo del proyecto es ser capaces de descargar los datos del calendario macro económico de la página <http://www.forexfactory.com/calendar>. Algunos de los datos que nos podremos descargar están publicados en las páginas web oficiales de cada país. Recolectar todos estos datos puede ser una tarea dura debido a que cada país tiene su propio formato, los datos no están organizados, etc. Es por esto, que nos será de gran utilidad ser capaces de tener todos los datos de una vez y posteriormente poder aplicarnos en nuestros análisis macro económicos.

## 1.2 Estructura de los datos

A continuación se muestra el formato del calendario de la página de Forexfactory. El calendario está dividido en varias columnas:

- Día
- Hora de la noticia / evento.
- Moneda a la que afectará el evento.
- Impacto: Se muestran de color amarillo aquellas noticias que son de poca relevancia, de color naranja aquellas que pueden tener un impacto medio y, finalmente, de color rojo aquellas que tienen un impacto importante.
- Breve descripción del tipo de noticia / comunicado.
- En la carpeta tenemos un enlace en el que tendremos algo más de información.
- Actual: Se muestra el valor de dato que se va a comunicar en el evento.
- Forecast: Previsión del dato que tiene que salir.
- Previous: Valor que tenía previamente el indicador en cuestión.

Tue  
May 23

12:30am

JPY

All Industries Activity m/m

-0.4%

0.7%

2:00am

CHF

Trade Balance

2.87B

3.10B

Specs

© Forex Factory

Source

Federal Statistical Office (latest release)

Measures

Difference in value between imported and exported goods during the reported month;

Usual Effect

Actual > Forecast = Good for currency;

Frequency

Released monthly, about 22 days after the month ends;

Next Release

Jun 22, 2017

FF Notes

A positive number indicates that more goods were exported than imported;

Why Traders Care

Export demand and currency demand are directly linked because foreigners must buy the domestic currency to pay for the nation's exports. Export demand also impacts production and prices at domestic manufacturers;

History

Actual

Forecast

Previous

Apr 27, 2017

3.10B

3.01B

3.12B

Mar 21, 2017

3.11B

3.85B

4.83B

Feb 21, 2017

4.73B

3.03B

2.69B

Jan 26, 2017

2.72B

2.81B

3.50B

Dec 20, 2016

3.64B

3.57B

2.66B

⌵ More

Graph

Related Stories

Currently no related stories. Submit Related Story

Figure 2: Detalles del evento

Si nos fijamos en la parte superior izquierda podemos ver el enlace **Next**. Este enlace nos lleva a la siguiente semana, por lo tanto para obtener los datos de diferentes días tendremos que ir de una página a la siguiente.

Además, si clicamos encima de la carpeta podemos obtener información detallada sobre el evento. Por ejemplo:

- Fuente de datos
- Explicación del evento
- Cómo suele afectar a las divisas que están relacionadas, es decir si el datos es positivo afecta bien o mal.
- Motivos por los que los especuladores (inversores) tienen en cuenta esta noticia.

## 2 Accediendo a los datos

Para poder acceder a los datos lo primero que tenemos que hacer es copiar la URL de un día en concreto y descarnos la página web con la función `read_html`. Una vez tengamos los datos accederemos a la tabla `.calendar_row` y lo guardaremos en la variable `web_selected`, que será una lista con todos los elementos.

```
url <- "https://www.forexfactory.com/calendar.php?day=may2.2017"
web_full <- read_html(url)
web_selected <- web_full %>% html_nodes("table .calendar_row")
```

Ahora que ya tenemos seleccionada la tabla general del calendario, tenemos que acceder a cada uno de los datos que nos interesa. Por ejemplo si queremos extraer la fecha accederemos indicando que en la primera posición de la lista `web_selected` seleccionaremos el nodo que tiene un tabla con la clase `.date`. Del mismo modo si queremos extraer la moneda que estará afectada le indicaremos que tiene que seleccionar el valor del nodo donde la clase sea `.currency`.

En el caso de los campos numéricos, como pueden ser *Actual*, *Forecast* y *Previous* se aplica una restricción para descartar caracteres que no nos interesan y quedarnos solamente con los números:












May 2, 2017							Up Next	Filter
Date	1:42pm	Currency	Impact	Detail	Actual	Forecast	Previous	Graph
Tue May 2	12:30am	AUD	 Cash Rate		1.50%	1.50%	1.50%	
		AUD	 RBA Rate Statement					
	1:00am	JPY	 BOJ Core CPI y/y		-0.1%	0.2%	0.1%	
	3:15am	EUR	 Spanish Manufacturing PMI		54.5	54.3	53.9	

Figure 3: Ejemplo de los datos

```
eventid="69729" data-touchable data-mousedown="true">
  <td class="calendar_cell calendar_date date">...</td>
  <td class="calendar_cell calendar_time time">All Day</td>
  <td class="calendar_cell calendar_currency currency">CNY</td>
  <td class="calendar_cell calendar_impact impact calendar_impact--holiday">
    <div class="calendar_impact-icon calendar_impact-icon--screen">
```

Figure 4: Código HTML

```
gsub("[^0-9&.-]", "", .)
```

y en el caso de los campos de texto eliminaremos los espacios extra del inicio y final de las frases:

```
trimws()
```

```
dates <- web_selected[1] %>%
  html_nodes(".date") %>%
  html_text()
```

```
currency <- web_selected[1] %>%
  html_nodes(".currency") %>%
  html_text() %>%
  trimws() %>%
  as.character()
```

```
actual_value <- web_selected[1] %>%
  html_nodes(".actual") %>%
  html_text() %>%
  gsub("[^0-9&.-]", "", .) %>%
  as.numeric()
```

```
data.frame("Date" = dates, "Currency" = currency, "Actual" = actual_value)
```

Date	Currency	Actual
TueMay 2	AUD	1.5

## 2.1 Extracción de los datos de toda la semana

Una vez tenemos todos los datos que queremos seleccionar crearemos una función para que el código sea más flexible.

```
extract_data_of_day <- function(current_day) {
```

```
  dates <- current_day %>%
    html_nodes(".date") %>%
    html_text()
```

```

    html_nodes(".currency") %>%
    html_text() %>%
    trimws() %>%
    as.character()

    impact <- current_day %>%
    html_nodes(".impact span") %>%
    html_attr("class") %>%
    trimws()

    event_name <- current_day %>%
    html_nodes(".event span") %>%
    html_text() %>%
    trimws()

    actual_value <- current_day %>%
    html_nodes(".actual") %>%
    html_text() %>%
    gsub("[^0-9&.-]", "", .) %>%
    as.numeric()

    forecast_value <- current_day %>%
    html_nodes(".forecast") %>%
    html_text() %>%
    gsub("[^0-9&.-]", "", .) %>%
    as.numeric()

    previous_value <- current_day %>%
    html_nodes(".previous") %>%
    html_text() %>%
    gsub("[^0-9&.-]", "", .) %>%
    as.numeric()

    res <- data.frame(
      "Currency" = currency,
      "Impact" = ifelse(length(impact) == 0, NA, as.character(impact)),
      "Event" = event_name,
      "Actual" = actual_value,
      "Forecast" = forecast_value,
      "Previous" = previous_value
    )
    return(res)
}

# Test Function
extract_data_of_day(web_selected) %>% cbind(date = "2017-05-02", .) %>% head(6)

```

date	Currency	Impact	Event	Actual	Forecast	Previous
2017-05-02	AUD	high	Cash Rate	1.5	1.5	1.5
2017-05-02	AUD	high	RBA Rate Statement	NA	NA	NA
2017-05-02	JPY	high	BOJ Core CPI y/y	-0.1	0.2	0.1
2017-05-02	EUR	high	Spanish Manufacturing PMI	54.5	54.3	53.9
2017-05-02	CHF	high	Manufacturing PMI	57.4	58.2	58.6
2017-05-02	EUR	high	Italian Manufacturing PMI	56.2	55.9	55.7

Con la función `extract_data_of_day` obtenemos toda la información de un día del calendario, pero qué pasa si queremos sacar la de todos los días de la semana que tenemos cargada? Para ello tenemos que iterar sobre cada uno de los días. Lo haremos con la siguiente línea de código:

```
web_filtered <- lapply(web_selected, extract_data_of_day) %>% bind_rows()
web_filtered %>% head()
```

Currency	Impact	Event	Actual	Forecast	Previous
AUD	high	Cash Rate	1.5	1.5	1.5
AUD	high	RBA Rate Statement	NA	NA	NA
JPY	low	BOJ Core CPI y/y	-0.1	0.2	0.1
EUR	medium	Spanish Manufacturing PMI	54.5	54.3	53.9
CHF	low	Manufacturing PMI	57.4	58.2	58.6
EUR	low	Italian Manufacturing PMI	56.2	55.9	55.7

## 2.2 Extracción de los datos de múltiples fechas

Hasta ahora hemos podido extraer toda la información de la semana que teníamos cargada en el navegador, pero nosotros lo que queremos es extraer todo el histórico de datos. Para ello tenemos que extraer los datos de todas las semanas anteriores y unirlos.

Si nos fijamos, la dirección URL está formada de la siguiente manera:

<http://www.forexfactory.com/calendar.php?day=may2.2017>. Lo que nos importa es **may2.2017**:

- may: Mes abreviado
- 2: día del mes
- 2017: Año

Sabiendo esto, podemos crear de forma automática las URL's de cada día y extraer los datos iterando sobre cada URL.

```
day <- as.Date("2017-05-02")

# Creamos las abreviaciones de cada mes
months_abbrev <- c("jan", "feb", "mar", "apr", "may", "jun",
                  "jul", "aug", "sep", "oct", "nov", "dec")

# Hacemos un split de la fecha en día, mes y año
day_of_month <- format(day, "%d") %>% as.integer()
```

```

number_of_month <- format(day,"%m") %>% as.integer()
number_of_month <- months_abbrev[number_of_month]
year <- format(as.Date(day), "%Y") %>% as.integer()

# Construimos la URL del día seleccionado
url <- paste0("http://www.forexfactory.com/calendar.php?day=",
              number_of_month, day_of_month, ".", year)

# Descargamos los datos
web_full <- read_html(url)

# Seleccionamos la tabla principal
web_selected <- web_full %>% html_nodes("table .calendar_row")

```

Ahora solo tendríamos que llamar a la función `extract_data_of_day` (con argumento `web_selected`) para extraer los datos del día seleccionado, tal y como hemos hecho anteriormente.

Igual que en el paso anterior crearemos una función para que el código sea más flexible y modular. La función solo tiene un argumento `range_dates`, un vector con las fechas que queremos extraer la información

```

get_macro_data <- function(range_dates) {

  # Lista para guardar la informacion de cada día
  dataset <- list()

  # Iteramos sobre cada día del vector de fechas
  for (day in range_dates) {
    day <- as.Date(day)
    cat("Downloading:", format(day, "%Y-%m-%d"), "\n")

    # Hacemos un split de la fecha en día, mes y año
    day_of_month <- format(day, "%d") %>% as.integer()
    number_of_month <- format(day,"%m") %>% as.integer()
    number_of_month <- months_abbrev[number_of_month]
    year <- format(as.Date(day), "%Y") %>% as.integer()

    # Construimos la URL.
    # Ejemplo: http://www.forexfactory.com/calendar.php?day=jan1.2007
    url <- paste0("http://www.forexfactory.com/calendar.php?day=",
                  number_of_month, day_of_month, ".", year)
    web_full <- read_html(url)
    web_selected <- web_full %>% html_nodes("table .calendar_row")

    # Seleccionamos la informacion
    web_filtered <-
      web_selected %>%
      lapply(extract_data_of_day) %>%

```



```

    bind_rows() %>%
    cbind(date = day, .)

    # Store all the data in a list
    dataset[[as.character(day)]] <- web_filtered
  }

  return(dataset)
}

```

## 2.3 Unión de todo lo anterior: Ejemplo completo

Con las dos funciones creadas ya podemos contruir las líneas de código que nos haran de *main*. En primer lugar indicamos la fecha de inicio y fin, y creamos la secuencia de fechas. Después definimos todos los meses abreviados con el formato que utiliza la fecha, aunque también se puede utilizar la función *month* del paquete *lubridate*, tal y como se muestra en los comentarios.

Finalmente llamamos a la funcion *get\_macro\_data*

```

# Define start and end dates
start_date <- as.Date("2017-05-28")
end_date   <- as.Date("2017-05-29")
range_dates <- seq(from = start_date, to = end_date, by = "1 day")

# Creamos las abreviaciones de cada mes
months_abbrev <- c("jan", "feb", "mar", "apr", "may", "jun",
                  "jul", "aug", "sep", "oct", "nov", "dec")

# Tambien podemos crearlo utilizando la function 'month' de la libreria lubridate
# months_abbrev <- lubridate::month(1:12, label = T)

# Run the script
t_start <- Sys.time()
data_raw <- get_macro_data(range_dates) %>%
  bind_rows() %>% filter(Currency != "") %>%
  mutate(Currency = as.factor(Currency), Impact = as.factor(Impact))

## Downloading: 2017-05-28
## Downloading: 2017-05-29

#saveRDS(data_raw, "data_raw.rds")
Sys.time() - t_start

```

```
## Time difference of 2.910407 secs
```

Vamos a comprobar que se hayan descargado los datos correctamente (las fechas no son las mismas que las que hemos indicado porque estamos cargando un fichero donde nos hemos descargado un rango más amplio)

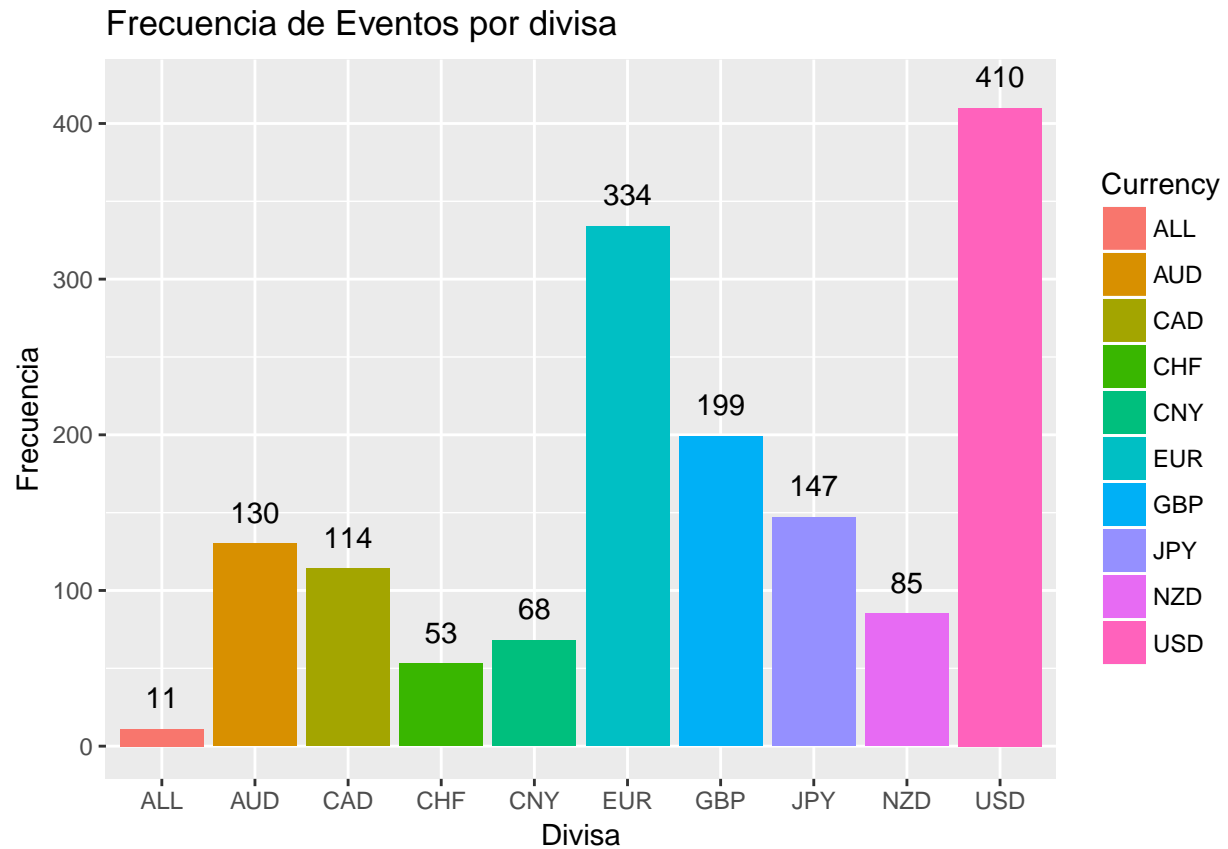
```
data_raw <- readRDS("data_raw.rds")
bind_rows(head(data_raw, 3), tail(data_raw, 3))
```

date	Currency	Impact	Event	Actual	Forecast	Previous
2017-01-29	NZD	medium	Trade Balance	-41.0	-95.0	-746.0
2017-01-29	JPY	low	Retail Sales y/y	0.6	1.6	1.7
2017-01-29	CNY	holiday	Bank Holiday	NA	NA	NA
2017-05-29	JPY	low	Retail Sales y/y	NA	2.2	2.1
2017-05-29	CNY	holiday	Bank Holiday	NA	NA	NA
2017-05-29	AUD	medium	Building Approvals m/m	NA	3.2	-13.4

### 3 Análisis de los datos

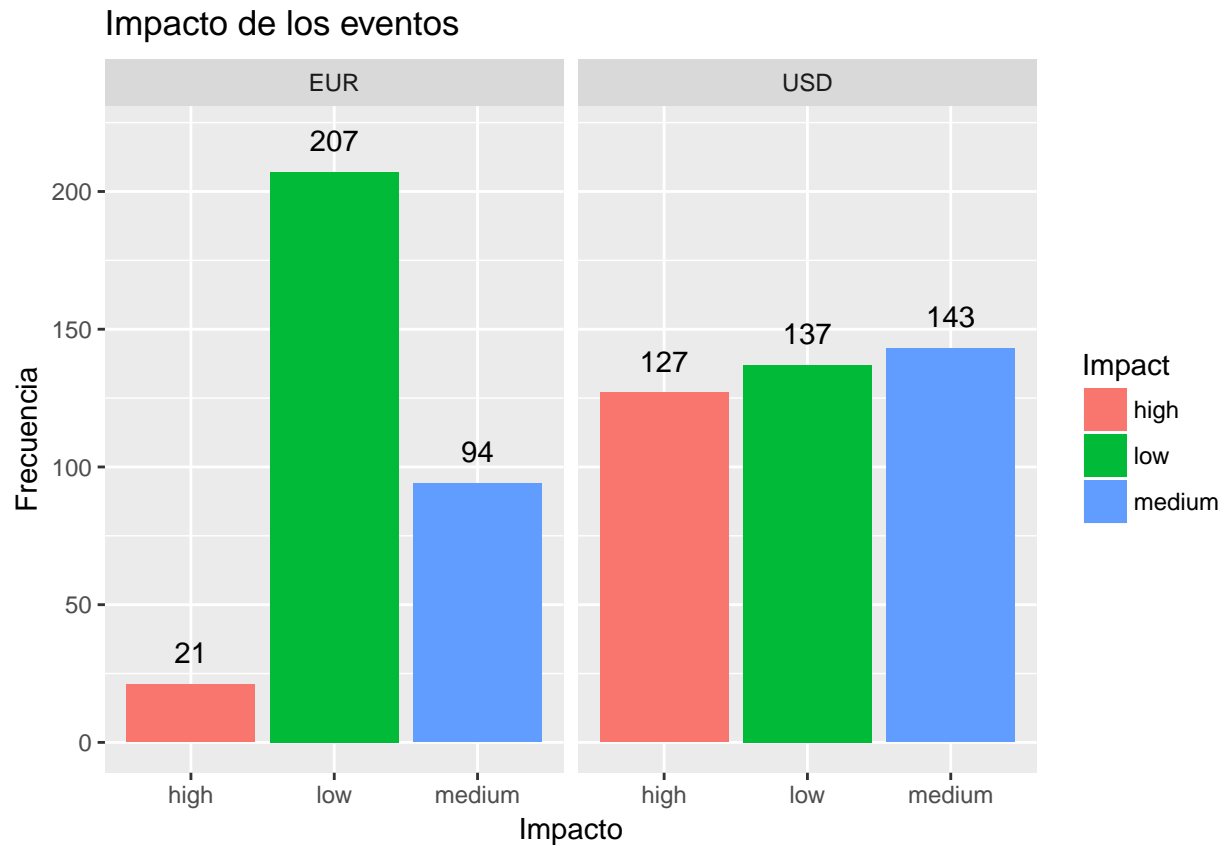
Con todos los datos descargados podemos hacer múltiples análisis, para empezar vamos a hacer un descriptivo de los datos para hacernos una idea principal de qué es lo que tenemos:

```
ggplot(data_raw, aes(x = Currency, fill = Currency)) +
  geom_bar(stat = "count") +
  geom_text(stat = "count", aes(label = ..count..), vjust = -1) +
  ggtitle("Frecuencia de Eventos por divisa") +
  xlab("Divisa") + ylab("Frecuencia") +
  ylim(0, 420)
```



Claramente tenemos muchos más eventos para el Euro y para el Dólar que para el resto de divisas. Vamos a ver de qué tipo son:

```
data_raw %>% filter(Currency %in% c("EUR", "USD") & Impact != "holiday") %>%
ggplot(aes(x = Impact, fill = Impact)) +
  geom_bar(stat = "count") +
  geom_text(stat = "count", aes(label = ..count..), vjust = -1) +
  facet_wrap(~ Currency) +
  ggtitle("Impacto de los eventos") +
  xlab("Impacto") + ylab("Frecuencia") +
  ylim(0, 220)
```



Podemos seleccionar aquellos que han salido más veces y vamos a crear un objeto de tipos *time series*.

```
data_raw %>%
  filter(Currency %in% c("EUR", "USD")) %>%
  group_by(Currency, Event) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  head(4)
```

Currency	Event	n
USD	Crude Oil Inventories	17
USD	Natural Gas Storage	17
USD	Unemployment Claims	17
EUR	ECB President Draghi Speaks	11

Una vez tenemos los datos en formato time series podemos aplicar todos los análisis que hemos aprendido en otras asignaturas.

```
COI <- data_raw %>%
  filter(Event == "Crude Oil Inventories")

COI_Actual <- ts(COI$Actual, start = c(2017, 05), frequency = 52)
```

```
COI_Forecast <- ts(COI$Forecast, start = c(2017, 05), frequency = 52)

ts.plot(COI_Actual, COI_Forecast, col = 1:2)
legend("topright", c("CIO Actual", "COI Forecast"), col = 1:2, lty = 1, cex = 0.9)
```

