Creating process models, reading process logs and analyzing process data is a key step in understanding and improving business processes. bupaR is an open-source suite for the handling and analysis of business process data, developed by the Business Informatics research group at Ghent University, Belgium. The central package includes basic functions for data analysis and visualization, and there are several additional packages that can be used to get information about an event log and also provide specific event log versions of generic functions.

### Getting Started

```r
install.packages("bupaR")
library(bupaR)
```

### Process Analysis Workflows

The central package provides functions to conduct process mining, both from a causal analysis perspective and with a focus on performance. The `bupa` function allows the user to analyze centrality of events and a performance perspective. The `bupa` function provides a set of driver-impact tools that can be used to measure the driver-impact of different interventions.

```r
bupa(eventlog, filter_method = "start", reverse = T)
```

### Process Dashboards

The `processmonitR` package provides functionality to monitor and visualize processes through the use of dashboards. Each dashboard provides a set of event log data that can be used in combination with the `bupa` function to visualize the data.

```r
processmonitR::process_monitR()
```

### Conditional Process Analysis

Conditional process analysis is a method to identify causal relationships within specific sub-processes. The `bupa` function allows the user to analyze conditional processes and identify causal relationships within specific sub-processes.

```r
conditional_process(eventlog, filter_method = "start", reverse = T)
```

### Reading and Writing XES-files

The `xesread` and `xeswrite` packages provide functionality to read and write XES-files. These packages can be used to read and write event logs in the XES format, which is an open standard for representing event logs.

```r
xesread(xesfile)
xeswrite(xesfile, eventlog)
```

### Event Data Subsetting

Event data subsetting can be done on two levels: on the level of events and on the level of traces. Each of the subsetting methods returns an event log that is a subset of the original event log.

```r
event_data_subsetting(eventlog, filter_method = "start", reverse = T)
```

### Data Mining

Data mining is the process of discovering patterns in large data sets, including data cubes and data trees. The `data Mining` package provides functionality to conduct data mining and analyze the results.

```r
dataMining::dataMining(xesfile)
```

### Exploratory and Descriptive Event Analysis

Exploratory and descriptive event analysis is a method to gain insights into the behavior of business processes. The analysis includes the calculation of various metrics and visualization of the results.

```r
exploratoryDescriptiveEventAnalysis(eventlog, filter_method = "all")
```

### Process Visualizations

Process visualizations provide a visual representation of the event log data. They can be used to gain insights into the behavior of business processes and identify patterns and trends.

```r
processVisualizations(eventlog, filter_method = "all")
```

### Summary

The `bupa` package provides a set of functions for the analysis and visualization of event logs. The package includes basic functions for data analysis and visualization, and there are several additional packages that can be used to get information about an event log and also provide specific event log versions of generic functions.