~~~~
<!--
%\VignetteEngine{knitr}
%\VignetteIndexEntry{Getting started}
-->

```{r, echo = FALSE, message = FALSE}
set.seed(26)
knitr::opts_chunk$set(
  comment = "#>",
  error = FALSE,
  tidy = FALSE,
  fig.width = 7,
  fig.height = 6)
```


# Getting started with REPRA

The Renewable Energy Probabilistic Resource Adequacy tool (REPRA) can be used to answer resource adequacy questions
in the presence of variable generation (VG). This document shows an example of the basic capabilities of the package.
The examples utilize part of the built-in datasets that are included in the package.

First load the package into memory, along with others that are used in this demonstration.

```{r, message = FALSE}
library(repra)
library(dplyr)
library(reshape2)
library(ggplot2)
```


## Input data

All the calculations require two basic inputs, which are introduced in this section: an outage table and time series
data.


### Outage table

An outage table combines existing conventional generation and indicates the loss-of-load probability (LOLP) for
different load levels. This object is created with the `outage_table` function, which requires input to be formatted
as a `data.frame` with a column of capacity and expected forced outage rates (EFOR). For more information on the
function, consult the help `?outage_table`.

```{r}
gens <- repragen %>%
  filter(Area == "AZ-NM-NV")
head(gens)
out.table <- outage_table(gens)
head(out.table)
```

An outage table can be easily plotted and summarized.

```{r}
plot(out.table)
summary(out.table)
```


### Time series data

Time series need to be formatted with the `format_timedata` function before they can be used by other functions in
the package. Time series must contain, at least, load data. Additional columns will be assumed to be VG and will be
automatically be used in the other calculations.

```{r}
tdata <- repratime %>% filter(Area == "AZ-NM-NV")
head(tdata)
td <- format_timedata(tdata)
head(td)
```


## Calculate metrics

To calculate reliability metrics, a time series with a `NetLoad` column must be provided to `calculate_metrics`. For
example, here are the metrics for load (in this case the load is scaled or otherwise all values would be zero).

```{r}
td2 <- td %>%
  mutate(NetLoad = 2.02 * Load)
calculate_metrics(td2, out.table)
```
~~~~

The `raw` parameter allows us to see the actual values of LOLP and EUE for each hour.

```{r}
cv <- calculate_metrics(td2, out.table, raw = TRUE)
cv %>%
  arrange(-Capacity) %>%
  as.data.frame() %>%
  head()
```

These metrics can also be calcuated after adding wind.

```{r}
td2.wind <- td %>%
  mutate(NetLoad = 2.02 * Load - Wind)
calculate_metrics(td2.wind, out.table)
```

Calculate ELCC

Effective load carrying capacity (ELCC) is the maximum load that can be served at a give reliability metric (by default 1 day in 10 years daily LOLE). For this calculations, the load shape is modified until the desired level is reached. For more information on this process consult the help page for `calculate_elcc`.

```{r}
elcc <- calculate_elcc(td, out.table)
as.data.frame(elcc)
```

By default, ELCC is calculated with all the available VG data. The following example takes the series `Wind` out of the calculations.

```{r}
elcc2 <- calculate_elcc(td, out.table, ignore = "Wind")
as.data.frame(elcc2)
```

ELCC can be calculated for different metrics and different objective values.

```{r}
elcc3 <- calculate_elcc(td, out.table, obj.metric = "LOLH", obj.value = 1.0)
as.data.frame(elcc3)
```

Calculate capacity value

Capacity value is the increase in ELCC (measured with the same risk level) when a new resource is added to the system. The `capacity_value` function automatically calculates capacity value for each of the VG time series and adds it to the load in order.

```{r}
cv <- capacity_value(td, out.table)
as.data.frame(cv)
```

This information can be reformatted and presented in a more pleseant format. It can also be easily plotted.

```{r}
dcast(cv, Level + Area + Metric + Objective ~ CVTech, value.var = "CV")
ggplot(cv, aes(x = paste(Level, Area, sep = ", "), weight = CV, fill = CVTech)) +
  geom_bar() +
  labs(x = "Area", y = "Capacity Value (MW)", fill = "Type")
```

The previous example calculated the cumulative capacity value by iteratively adding VG profiles. An alternative would be to calculate the capacity value by removing only one VG profile from the total mix. The parameter `marginal` is used to change this behavior.

```{r}
cv.mar <- capacity_value(td, out.table, marginal = TRUE)
as.data.frame(cv.mar)
```