

PV - Wirkungsgrad

Vorbereitungen

Laden der nötigen Bibliotheken.

```
source("01-Bibliotheken-laden.R")
```

Zur Auswertung werden einige Funktionen benötigt, die hier definiert werden.

```
# rolling functions bilden und anhängen
source("02-Funktionen-bilden.R")
# -----
```

Einlesen der Dateien "Daten_dd_mm_yyyy.csv":

Die Daten liegen tageweise vor, sie enthalten Datensätze (Zeilen), die im 5-Minuten-Rhythmus erfasst werden. Beim Einlesen werden sie zusammengefügt. Spalten: leistung.pv ,leistung.stp, netzeinspeisung, netzbezug, batt_ladung, batt_entladung, ladezustand

Die beiden ersten Werte sind identisch, deswegen wird "leistung.stp" sofort gelöscht.

Alle in der Einheit W, mit Ausnahme von 'ladezustand', dieser wird beim Lesen als Prozentsatz übergeben und anschließend auf 10000 = 100% normiert weil die Batterie eine Kapazität von annähernd 10kWh besitzt, kann dies auch als Wh gelesen werden. Die Zeilen müssen sortiert werden, weil die Dateien nicht in der korrekten zeitlichen Reihenfolge eingelesen werden.

```
# Einlesen der Datenfiles-----benoetigt rolling functions-----
source("03-Files-einlesen.R")
```

```
## gelesen:
```

```
## Daten_01_10_2017.csv Daten_01_11_2017.csv Daten_02_10_2017.csv Daten_02_11_2017.csv Daten_03_10_2017.csv
```

```
#-----
```

Zur Vorbereitung werden einige Spalten gelöscht, ergänzt, einige neu normiert:

- Die neue Spalte 'ct' zählt die Datenzeilen
- Über 'ladezustand' laeuft eine Glaettungsfunktion, um einzelne Ausfaelle in den Messungen zu beseitigen.
- 'day' und 'hour' werden aus der Variablen 'zeit' extrahiert und im Datumsformat "yyyy-mm-dd" bzw. als Zahl 0 - 23 gespeichert.
- 'ladediff' wird als Differenz von Ladezustand zwischen dem aktuellen Zustand und dem vorangegangenen berechnet (Einheit Wh).
- 'batt_ladung' und 'batt_entladung' werden von W in Wh umgerechnet (W in der Zeit 5 min, deswegen Division durch 12)

Anm.: In der späteren Auswertung wird dies so interpretiert: Eine zur Zeit t erbrachte Leistung P führt zu einer el. Arbeit von P.5min im Zeitintervall t plus/minus 2,5 min

```
# Ergaenzende Spaltenoperationen -----
source("03-Spalten-bearbeiten.R")
```

```
## Loesche Spalten leistung.pv netzeinspeisung netzbezug .
## Der Datensatz enthaelt jetzt 14393 Zeilen.
```

```
# -----
```

Festlegen von Arbeitsdaten und Initialisierung weiterer Variablen

```
# Arbeitsdaten festlegen
proj_level <- tibble() # Initialisierung einer Variablen, die zur Erzeugung von Plots dient
```

“level” legt die Höhe des Ladezustands fest, der als Basis für die Berechnung von Wirkungsgraden dient.

```
level <- 5000 # ..%-Level
```

Auswertungen

Tägliche Minima und Maxima identifizieren - optional

```
# Minima und Maxima markieren frueher R_Min_Max_mark.R
source("04_Auswertungen_Min_Max_tgl_Per.R")
#-----
```

Perioden zwischen horizontalen Niveaus bilden

```
# NEU_ Perioden zwischen horizontalen Niveaus bilden NEU form_level_pd.R
source("04_Auswertungen_gen_Per_zu_horiz_Niv.R")
#-----
# Ladung und Entladung nach täglichen MinMaxperioden bzw Level-Halbperioden summieren Lade_Entlade_Sum
source("04_Auswertungen_Summ_Ent_Ladung_in_Level_Halbper.R")
#-----
```

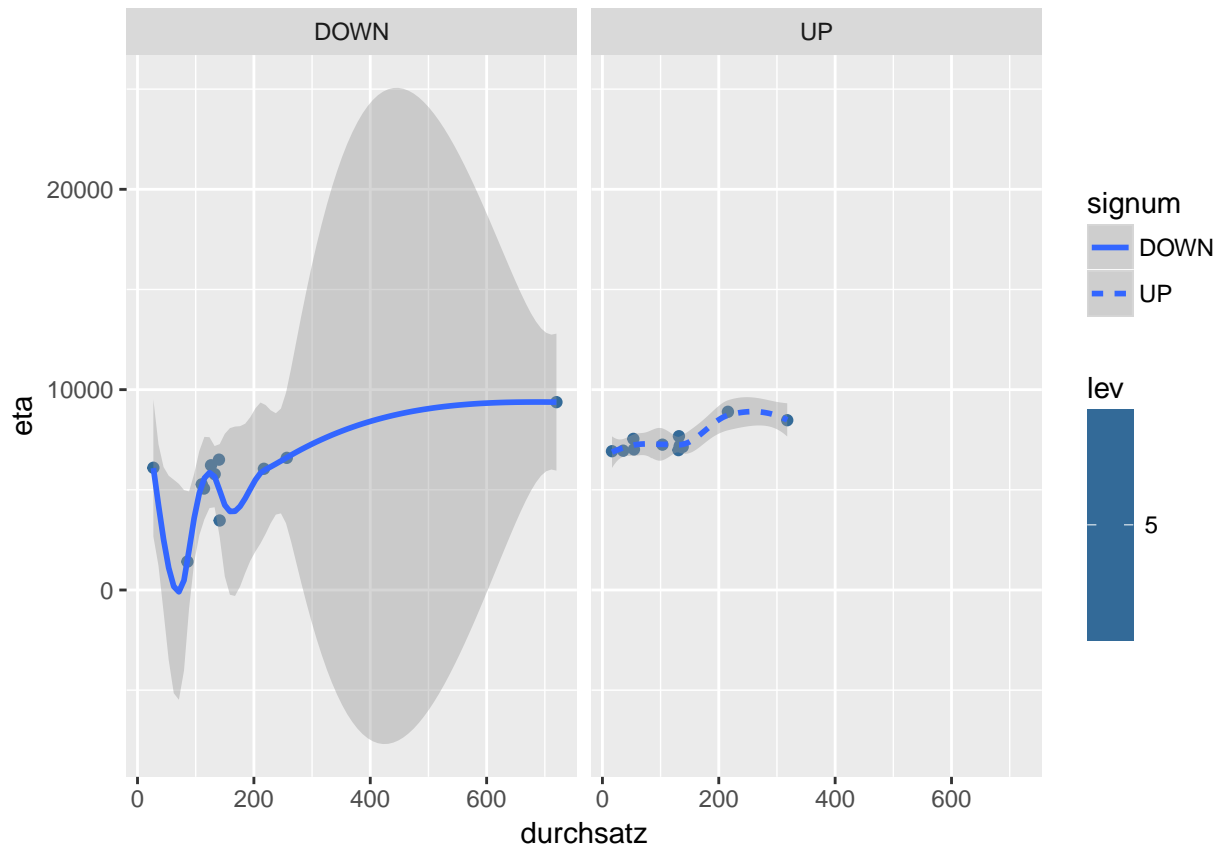
Graphische Auswertungen

Darstellung der Wirkungsgrade in Abhängigkeit von Durchsatz

Die maximale Energiedifferenz zwischen höchsten und niedrigsten Wert im Speicher in einer Halbperiode dividiert durch die Dauer der Halbperiode wird als (Energie-)Durchsatz bezeichnet angegeben in Wh/h. ‘lev’ bezeichnet dem gewählten level dividiert durch 1000.

```
#----- Proj_Ergeb_Level.R
source("05_Grafik_Vs_Durchsatz.R", print.eval=TRUE)
```

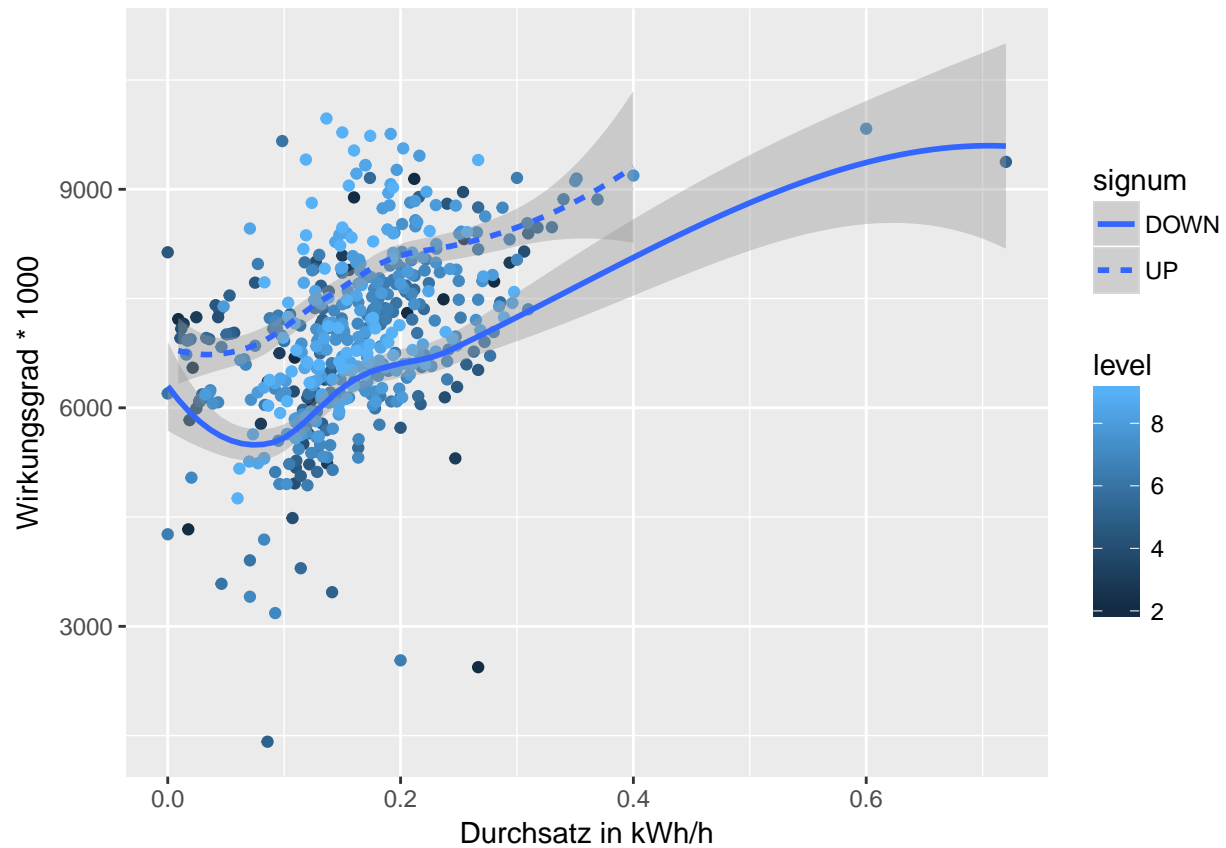
```
## `geom_smooth()` using method = 'loess'
```



Durch Wahl mehrerer Levelwerte erhält man folgende Darstellung

```
some_levels <- c(2000, 3000, 4000, 4500, 5000, 5500, 6000, 6500, 7000, 7500, 8000, 8500, 9000)
proj_level <- tibble()
for ( i in some_levels) { # zu jedem level den Datensatz auswerten und reduzieren auf einen Wert pro P
  level <- i
  source("04_Auswertungen_gen_Per_zu_horiz_Niv.R")
  #-----
  # Ladung und Entladung nach täglichen MinMaxperioden bzw Level-Halbperioden summieren
  source("04_Auswertungen_Summ_Ent_Ladung_in_Level_Halbper.R")
  #-----
  # Einen Datensatz reduzieren auf je einen Satz pro levelpd
  source("04_Auswertungen_Proj_1_wert_pro_levelpd.R")
  #-----
  proj_level <- rbind(proj_level, proj_level_new)
}
proj_level %>%
  filter(eta != 0) %>%
  filter(eta <= 10000) %>%
  ggplot(aes(x = durchsatz/1000, y = eta)) +
  geom_point(aes(x = durchsatz/1000, y = eta, color=lev)) +
  geom_smooth(mapping = aes(x = durchsatz/1000, y = eta, linetype = signum )) +
  labs(
    x = "Durchsatz in kWh/h",
    y = "Wirkungsgrad * 1000",
    color = "level"
  )
##+
```

```
## `geom_smooth()`` using method = 'loess'
```

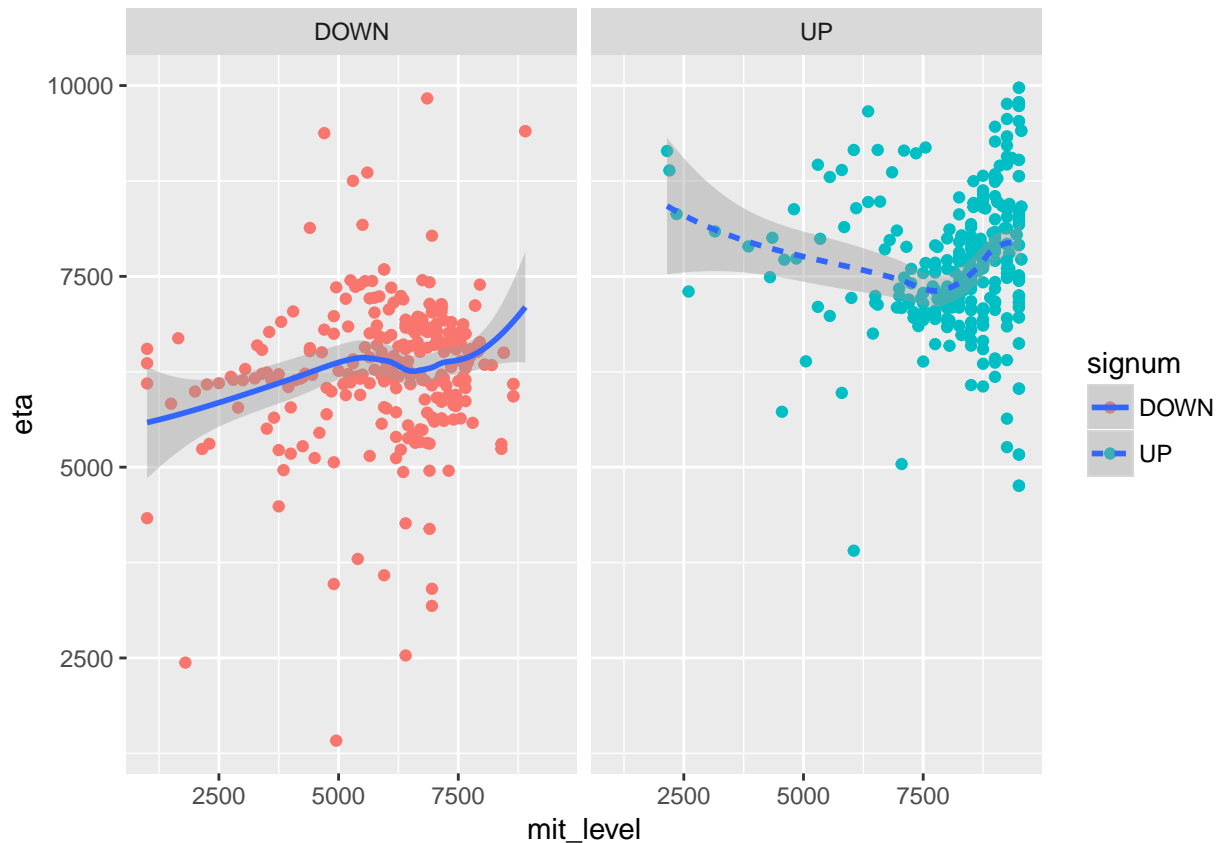


```
#geom_line(mapping = aes(x = durchsatz, y = eta, linetype = signum )) #+
#facet_wrap(~ level)
```

Darstellung der Wirkungsgrade in Abhängigkeit von der Mitte der Halbperiode

```
# ----- Proj_Level_Eta_vs_mit_level.R
source("05_Grafik_Eta_vs_Mitte.R", print.eval=TRUE)
```

```
## `geom_smooth()`` using method = 'loess'
```



Das Gleiche mit Aufsammlen von Daten zu mehreren Levels

Wirkungsgrade $\eta=0$ oder $\eta > 10000$ werden ausgeblendet.

```
some_levels <- c(1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000, 6500, 7000, 7500, 8000,
proj_level <- tibble()
for (level in some_levels) {
  # Perioden zwischen horizontalen Niveaus bilden
  source("04_Auswertungen_gen_Per_zu_horiz_Niv.R")
  #-----
  # Ladung und Entladung nach täglichen MinMaxperioden bzw Level-Halbperioden summieren
  source("04_Auswertungen_Summ_Ent_Ladung_in_Level_Halbper.R")
  #-----
  # Einen Datensatz reduzieren auf je einen Satz pro levelpd
  source("04_Auswertungen_Proj_1_wert_pro_levelpd.R")
  #-----
  proj_level <- rbind(proj_level, proj_level_new)
}
proj_level %>%
  filter(eta != 0 & eta <= 10000) %>%
  ggplot(aes(x = mit_level, y = eta)) +
  geom_point(aes(x = mit_level, y = eta, color=lev)) +
  geom_smooth(mapping = aes(x = mit_level, y = eta, linetype = signum )) +
  #geom_line(mapping = aes(x = durchsatz, y = eta, linetype = signum )) +
  facet_wrap(~ signum)
```

```
## `geom_smooth()` using method = 'loess'
```

