

PV - Wirkungsgrad

Geladen werden die Daten, die von der PV-Anlage mit Hilfe des SMA-Portals gewonnen werden. Ziel ist es, in der Tabelle *data* alle Datensätze aus den SMA-Daten, ergänzt um Hilfsgrößen, zur Verfügung zu stellen.

Die Daten liegen in Dateien tageweise vor, beim Download werden diese von Hand benannt, sie enthalten Datensätze (Zeilen), die im 5-Minuten-Rhythmus erfasst wurden. Beim Einlesen werden sie zusammengefügt.

Die Datensätze enthalten die Größen

leistung.pv — leistung.stp — netzeinspeisung — netzbezug — batt_ladung — batt_entladung — ladezustand

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

Vorbereitungen

Laden der nötigen Bibliotheken.

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

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

```
source("02-Funktionen-bilden.R")
```

Einlesen der Dateien "Daten_dd_mm_yyyy.csv":

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-----notig: 02-Funktionen-bilden.R-----  
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
```

Einige Spalten werden erzeugt, gelöscht, bearbeitet und z.B. 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 $P \cdot 5\text{min}$ im Zeitintervall $t \pm 2,5\text{min}$

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

```
## Loesche Spalten leistung.pv netzeinspeisung netzbezug .
## Der Datensatz enthaelt jetzt 14681 Zeilen.
```

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")
#-----
```

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

```
## 04_Auswertungen_Summ_Ent_Ladung_in_Level_Halbper.R
```

Perioden zwischen horizontalen Niveaus bilden - Neutrale Zyklen

Ein *neutraler Zyklus* ist eine Lade-Entlade-Vorgang der von einem Ladezustand ausgehend zu diesem zurückkehrt. Für diese ist es sinnvoll Wirkungsgrade als Verhältnis von Output zu Input zu bilden.

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

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

```
# NEU_ Perioden zwischen horizontalen Niveaus bilden NEU form_level_pd.R
source("04_Auswertungen_gen_Per_zu_horiz_Niv.R")
```

```
## 04_Auswertungen_gen_Per_zu_horizontNiv 5000
```

```
#-----
# Ladung und Entladung nach täglichen MinMaxperioden bzw Level-Halbperioden summieren Lade_Entlade_Sum_
source("04_Auswertungen_Summ_Ent_Ladung_in_Level_Halbper.R")
#-----
```

Zusammenfassung dieses Vorgangs

Dazu wird folgende Funktion definiert

```
zyklen_bilden <- function(x) {
  cat("zyklen bilden", i, "\n")
  level <- x
  source("04_Auswertungen_gen_Per_zu_horiz_Niv.R", local=TRUE)
  source("04_Auswertungen_Summ_Ent_Ladung_in_Level_Halbper.R", local=TRUE)
  source("04_Auswertungen_Proj_1_wert_pro_levelpd.R", local=TRUE)
  return(proj_level_new)
}
```

Sie gibt die Auswertung zurück mit jeweils einem Wert pro Zyklus

Graphische Auswertungen

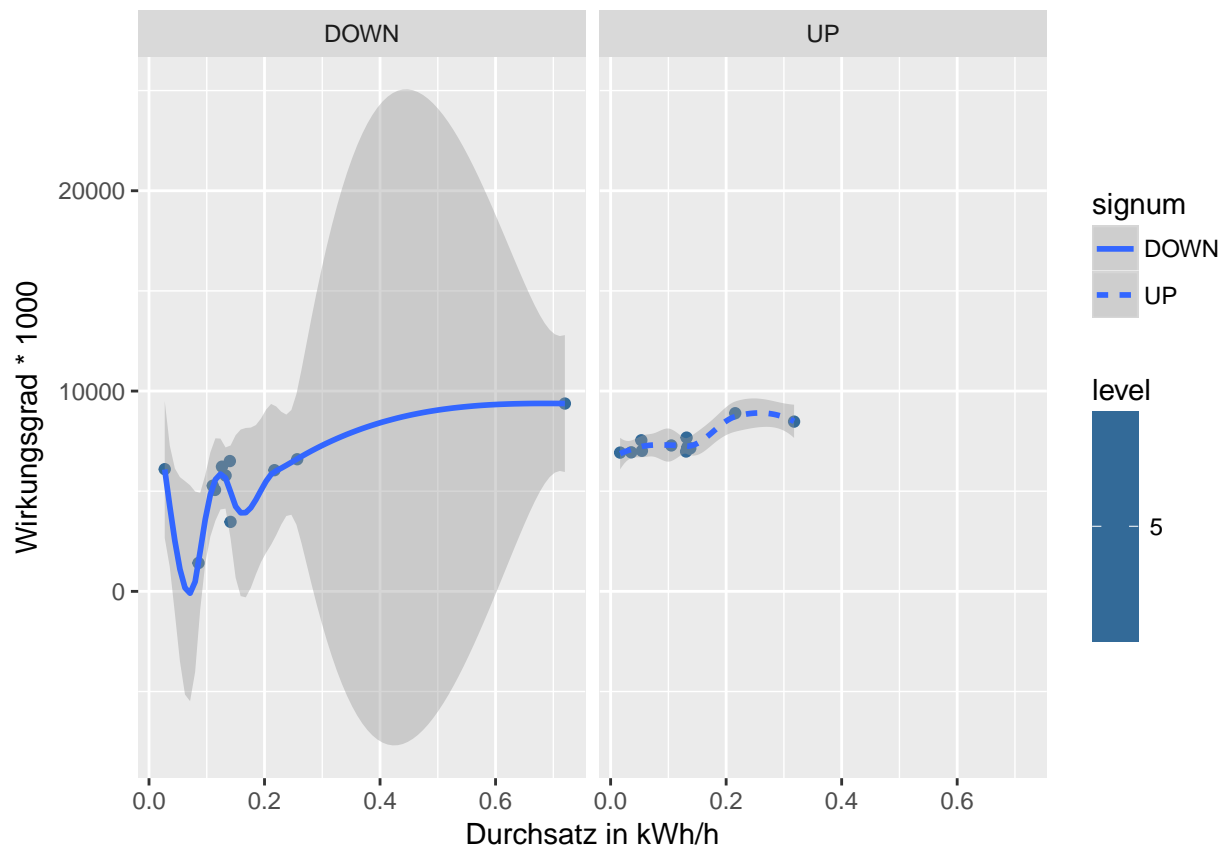
Darstellung der Wirkungsgrade in Abhängigkeit von Durchsatz

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)
```

```
## 05_Grafik_Vs_Durchsatz 5000
```

```
## `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  
  proj_level_new <- zyklen_bilden(i)  
  proj_level <- rbind(proj_level, proj_level_new)  
}
```

```
## zyklen bilden 2000  
## 04_Auswertungen_gen_Per_zu_horizontNiv 2000  
## zyklen bilden 3000  
## 04_Auswertungen_gen_Per_zu_horizontNiv 3000
```

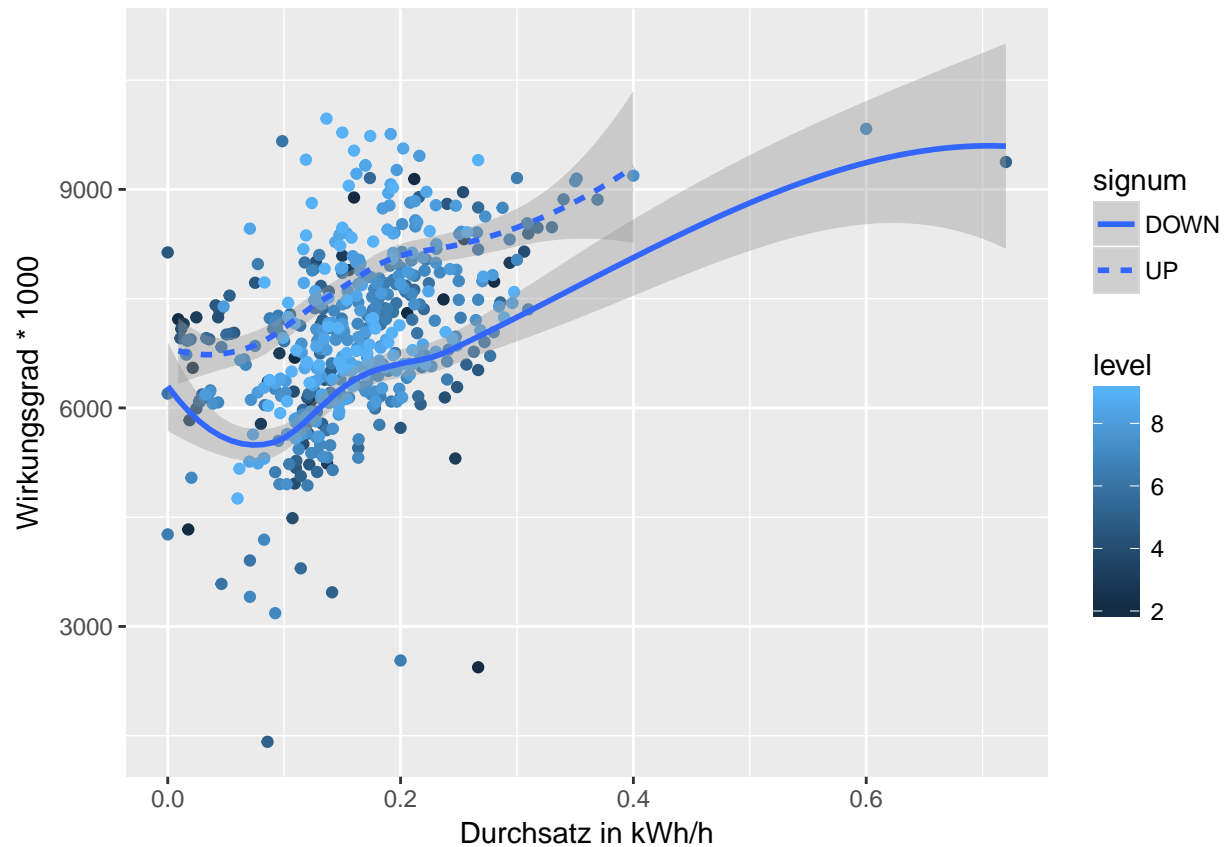
```

## zyklen bilden 4000
## 04_Auswertungen_gen_Per_zu_horizontNiv 4000
## zyklen bilden 4500
## 04_Auswertungen_gen_Per_zu_horizontNiv 4500
## zyklen bilden 5000
## 04_Auswertungen_gen_Per_zu_horizontNiv 5000
## zyklen bilden 5500
## 04_Auswertungen_gen_Per_zu_horizontNiv 5500
## zyklen bilden 6000
## 04_Auswertungen_gen_Per_zu_horizontNiv 6000
## zyklen bilden 6500
## 04_Auswertungen_gen_Per_zu_horizontNiv 6500
## zyklen bilden 7000
## 04_Auswertungen_gen_Per_zu_horizontNiv 7000
## zyklen bilden 7500
## 04_Auswertungen_gen_Per_zu_horizontNiv 7500
## zyklen bilden 8000
## 04_Auswertungen_gen_Per_zu_horizontNiv 8000
## zyklen bilden 8500
## 04_Auswertungen_gen_Per_zu_horizontNiv 8500
## zyklen bilden 9000
## 04_Auswertungen_gen_Per_zu_horizontNiv 9000

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'

```

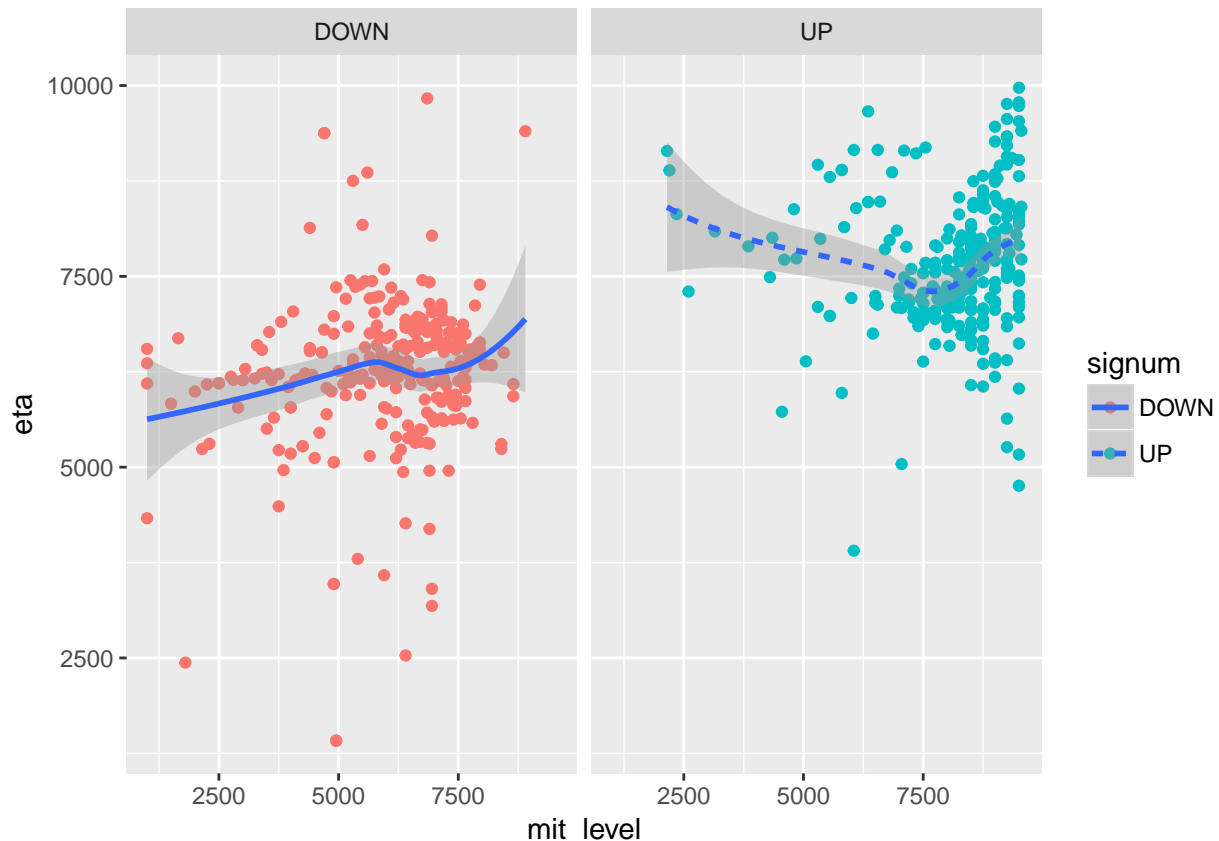


```
#geom_line(mapping = aes(x = durchsatz, y = eta, linetype = signum )) #+
#facet_wrap(~ level)
```

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 Aufsummieren 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)
}

## 04_Auswertungen_gen_Per_zu_horizontNiv 1000
## 04_Auswertungen_gen_Per_zu_horizontNiv 1500
## 04_Auswertungen_gen_Per_zu_horizontNiv 2000
## 04_Auswertungen_gen_Per_zu_horizontNiv 2500
## 04_Auswertungen_gen_Per_zu_horizontNiv 3000
## 04_Auswertungen_gen_Per_zu_horizontNiv 3500
## 04_Auswertungen_gen_Per_zu_horizontNiv 4000
```

```
## 04_Auswertungen_gen_Per_zu_horizontNiv 4500
## 04_Auswertungen_gen_Per_zu_horizontNiv 5000
## 04_Auswertungen_gen_Per_zu_horizontNiv 5500
## 04_Auswertungen_gen_Per_zu_horizontNiv 6000
## 04_Auswertungen_gen_Per_zu_horizontNiv 6500
## 04_Auswertungen_gen_Per_zu_horizontNiv 7000
## 04_Auswertungen_gen_Per_zu_horizontNiv 7500
## 04_Auswertungen_gen_Per_zu_horizontNiv 8000
## 04_Auswertungen_gen_Per_zu_horizontNiv 8500
## 04_Auswertungen_gen_Per_zu_horizontNiv 9000
```

```
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'
```

