

PV - Datenanalyse

01 Januar 2018

Contents

1	Präliminarien	1
2	Technische Vorbereitungen	2
2.0.1	Laden der nötigen Bibliotheken und Definition von Funktionen.	2
2.0.2	Einlesen der Dateien "Daten_dd_mm_yyyy.csv":	2
2.0.3	Ergänzende Spaltenoperationen und Datenkontrollen	2
3	Auswertungen	3
3.1	Einfache Summenbildungen	3
3.1.1	Wochen- und Monatssummen	3
3.1.2	Verbrauch und Erzeugung - Datenübersicht	3
3.1.3	Batteriezustand	4
3.1.4	Abschätzung von Wirkungsgraden für die Ladung bzw. die Entladung der Batterie . .	5
3.1.5	Visualisierung der Approximationsqualität	6
3.2	Strecken monotoner Entladung	7
3.3	Tägliche Minima und Maxima identifizieren - optional	7
3.4	Perioden zwischen horizontalen Niveaus bilden - Neutrale Zyklen	7
3.4.1	Bildung der Grundfunktionen	7
3.4.2	Zusammenfassung dieses Vorgangs	8
4	Graphische Auswertungen	8
4.1	Darstellung der Wirkungsgrade in Abhängigkeit von Durchsatz	8
4.2	Darstellung der Wirkungsgrade in Abhängigkeit von der Mitte der Halbperiode	10
4.2.1	Das Gleiche mit Aufsummeln von Daten zu mehreren Levels	11

1 Präliminarien

Die Programmstücke dienen der Auswertung der Daten, die von der PV-Anlage erzeugt und mit Hilfe des SMA-Portals gelesen werden.

Die Daten liegen tageweise in Dateien 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

zeit — leistung.pv — leistung.stp — netzeinspeisung — netzbezug — batt_ladung — batt_entladung — ladezustand(%)

Einheiten:

- Zeit als Datum-Stunde-Min-Sek
- Alle anderen Größen in W außer
- Ladezustand in % der Batteriekapazität. Laut Herstellerangabe beträgt diese 9,8 kWh, von denen 9,3 kWh verfügbar sind.

2 Technische Vorbereitungen

Dient nur der Darstellung des Codes, kann überlesen werden, wenn es nur um Anlagendaten geht.

2.0.1 Laden der nötigen Bibliotheken und Definition von Funktionen.

2.0.2 Einlesen der Dateien “Daten_dd_mm_yyyy.csv”:

Für jeden Tag liegt eine Datei vor, mit Messwerten im 5-min-Abstand. 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 10 kWh besitzt, kann die Zahl 10000 grob mit der Einheit Wh gelesen werden.

Beim Einlesen werden die Kopfzeilen und die erste, sonst leere Datenzeile für 0:00 Uhr ignoriert. Für jeden Tag gibt es also 288 Zeilen, endend mit 0:00 der Folgetages (diese Zeile wird aufgenommen).

Die Zeilen müssen sortiert werden, weil die Dateien nicht in der korrekten zeitlichen Reihenfolge eingelesen werden.

```
## Kontrolle der Kopfzeilen auf Gleichheit.  
## Die Kopfzeilen aller Dateien stimmen ueberein.  
##  
## Kontrolle auf redundante Zeilen:  
## Ok, es gibt keine doppelte Zeiten.
```

2.0.3 Ergänzende Spaltenoperationen und Datenkontrollen

- a) Die neue Spalte ‘ct’ zählt die Datenzeilen
- b) Über ‘ladezustand’ läuft eine Glättungsfunktion, um einzelne Ausfälle in den Messungen zu beseitigen.
- c) ‘month’, ‘day’ und ‘hour’ werden aus der Variablen ‘zeit’ extrahiert und im Datumsformat “yyyy-mm-dd” bzw. als Zahl 0 - 23 gespeichert.
- d) ‘ladediff’ wird als Differenz zwischen dem aktuellen Zustand und dem vorangegangenen Ladezustand berechnet.
- e) von W in Wh werden umgerechnet:
 - i) batt_ladung, batt_entladung,
 - ii) leistung.pv, leistung.stp, (bei der bestehende Anlage identisch)
 - iii) netzeinspeisung, netzbezug

Weil die Leistung jeweils zur Zeit t in 5min-Intervallen erhoben wird, wird in jedem Intervall die Arbeit $W = P \cdot 5\text{min} = P \cdot (1/12) \text{ h}$ verrichtet. Anm.: In der späteren Auswertung wird dies so interpretiert: Diese el. Arbeit von $P \cdot 5\text{min}$ wird im Zeitintervall $t \pm 2,5\text{min}$ erbracht

```
## Oct 2017 hat Daten von 31 Tagen  
## Nov 2017 hat Daten von 30 Tagen  
## Dec 2017 hat Daten von 31 Tagen  
## Jan 2018 hat Daten von 1 Tagen  
##  
## Warnung: An folgenden Tage liegen weniger als 288 Beobachtungen vor:  
## 2017-10-01 hat nur 280 Beobachtungen.  
## 2018-01-01 hat nur 1 Beobachtungen.  
##  
## Info: In 1027 Zeilen der Originaldaten findet gleichzeitig Ladung und Entladung statt.  
##  
## Der Datensatz enthaelt jetzt 26489 Zeilen.  
##
```

```
## Erzeuge Tabelle verbrauch
```

```
## Loesche aus data Spalten:  leistung.pv leistung.stp netzeinspeisung netzbezug .
```

Die Rohdaten werden gespalten in **data** mit den Variablen: zeit, batt_ladung, batt_entladung, ladezustand, month, week, day, hour, ladediff, ct

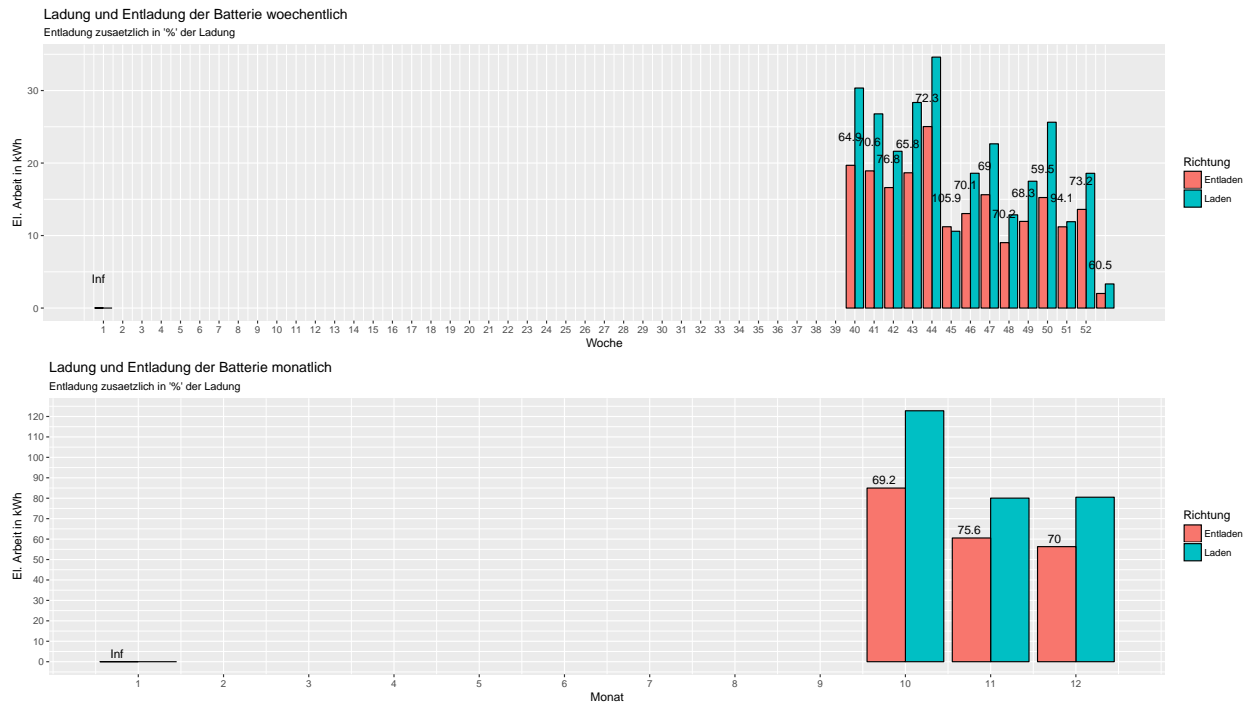
sowie **verbrauch** mit den Variablen: zeit, leistung.pv, netzeinspeisung, netzbezug, batt_ladung, batt_entladung, ladezustand, month, week, day, hour, ladediff, ct

3 Auswertungen

3.1 Einfache Summenbildungen

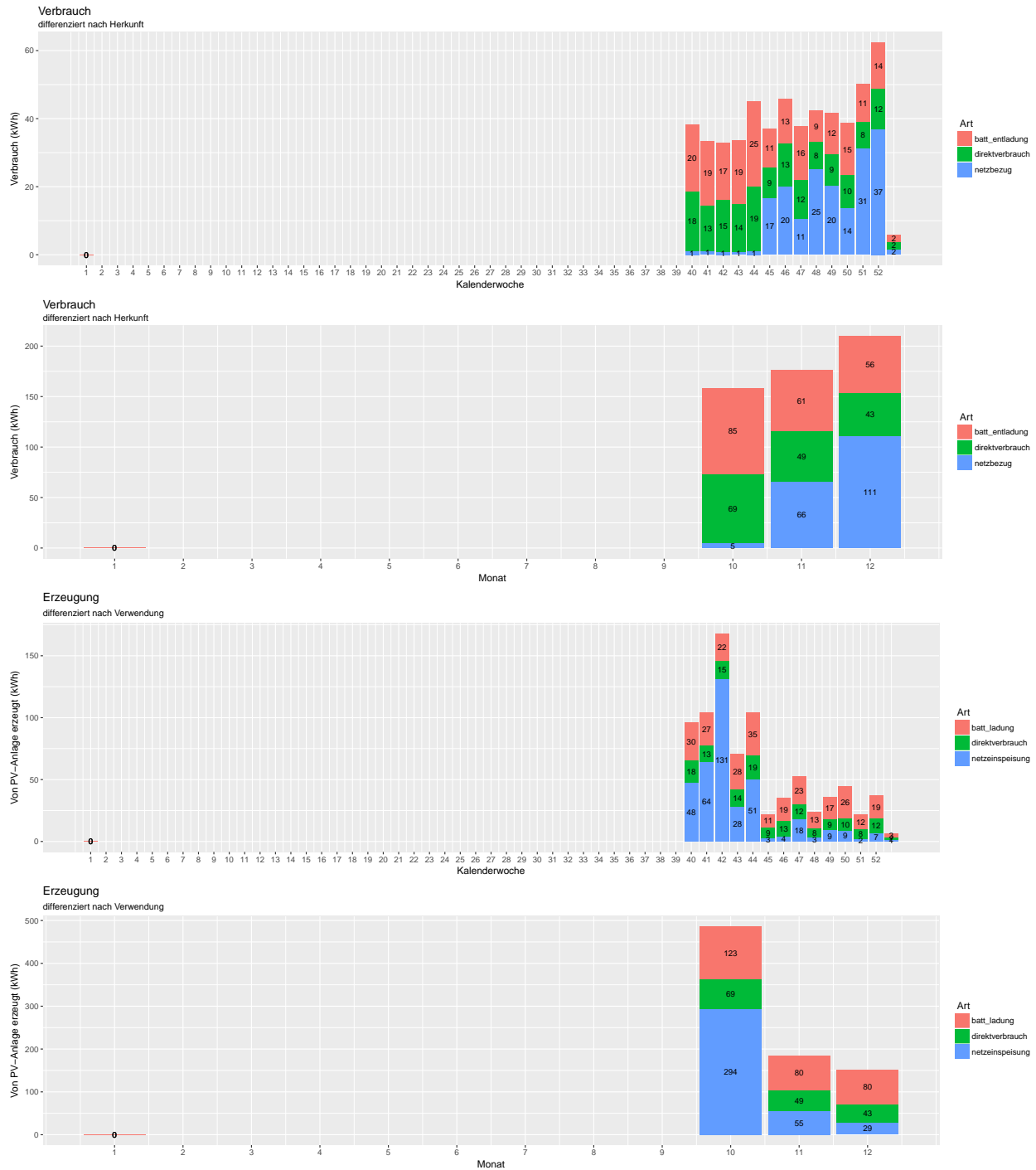
3.1.1 Wochen- und Monatssummen

Die folgenden Diagramme fassen die Lade- und Entladevorgänge der Batterie zusammen. Die daraus berechneten Wirkungsgrade sind gerade bei wöchentlicher Berechnung allenfalls Anhaltspunkte, weil der Batteriezustand zu Anfang bzw. zum Ende der Woche i.d.R. nicht gleich sein wird.



3.1.2 Verbrauch und Erzeugung - Datenübersicht

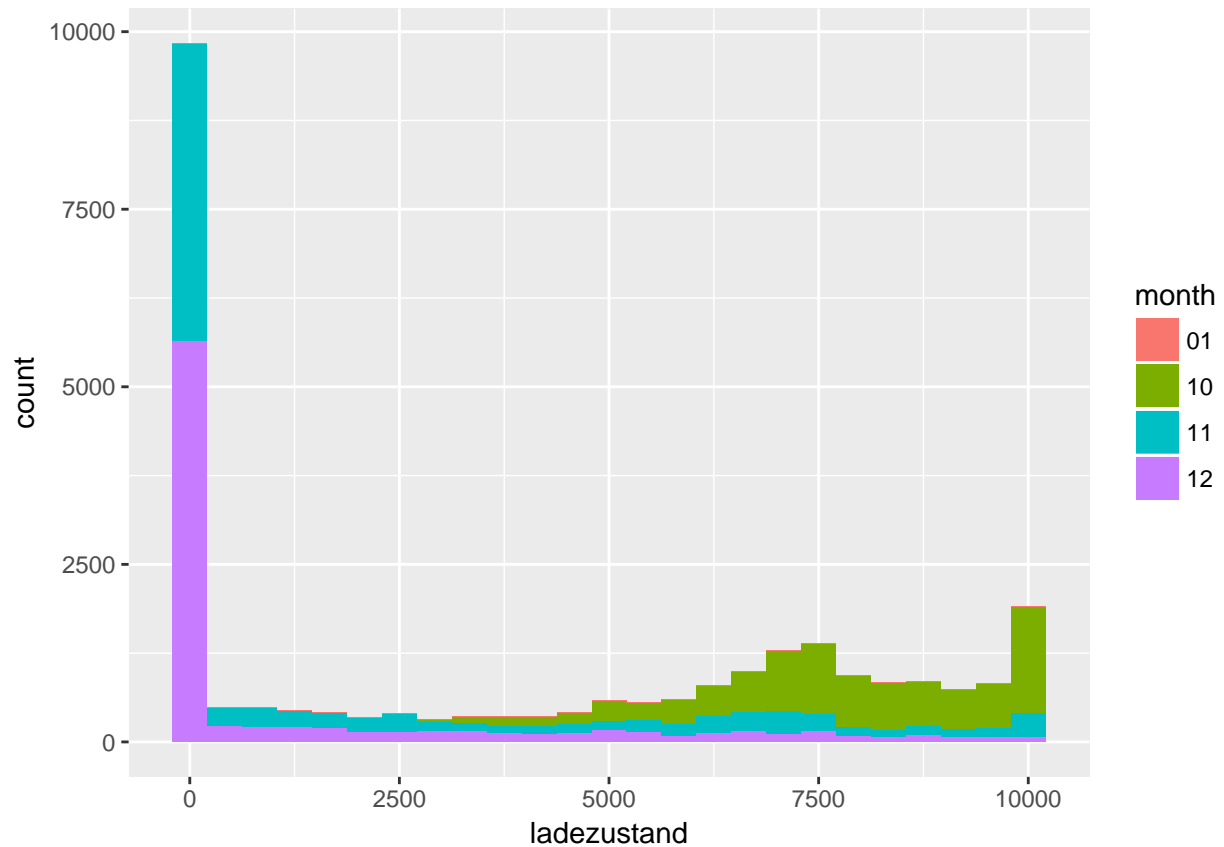
Die folgenden Diagramme haben die selbe Struktur, wie entsprechende Seiten der Website <http://sunnyportal.com/>.



3.1.3 Batteriezustand

Das folgende Diagramm zählt die Häufigkeit, mit der die Ladezustände der Batterie in den Monaten aufgetreten sind. Ausgezählt wird in Intervallen der Breite 0,400 kWh. (Amm: Ränder prüfen!)

Warning: Ignoring unknown aesthetics: order



Ladung und Entladung summieren und mit Ladezustand vergleichen

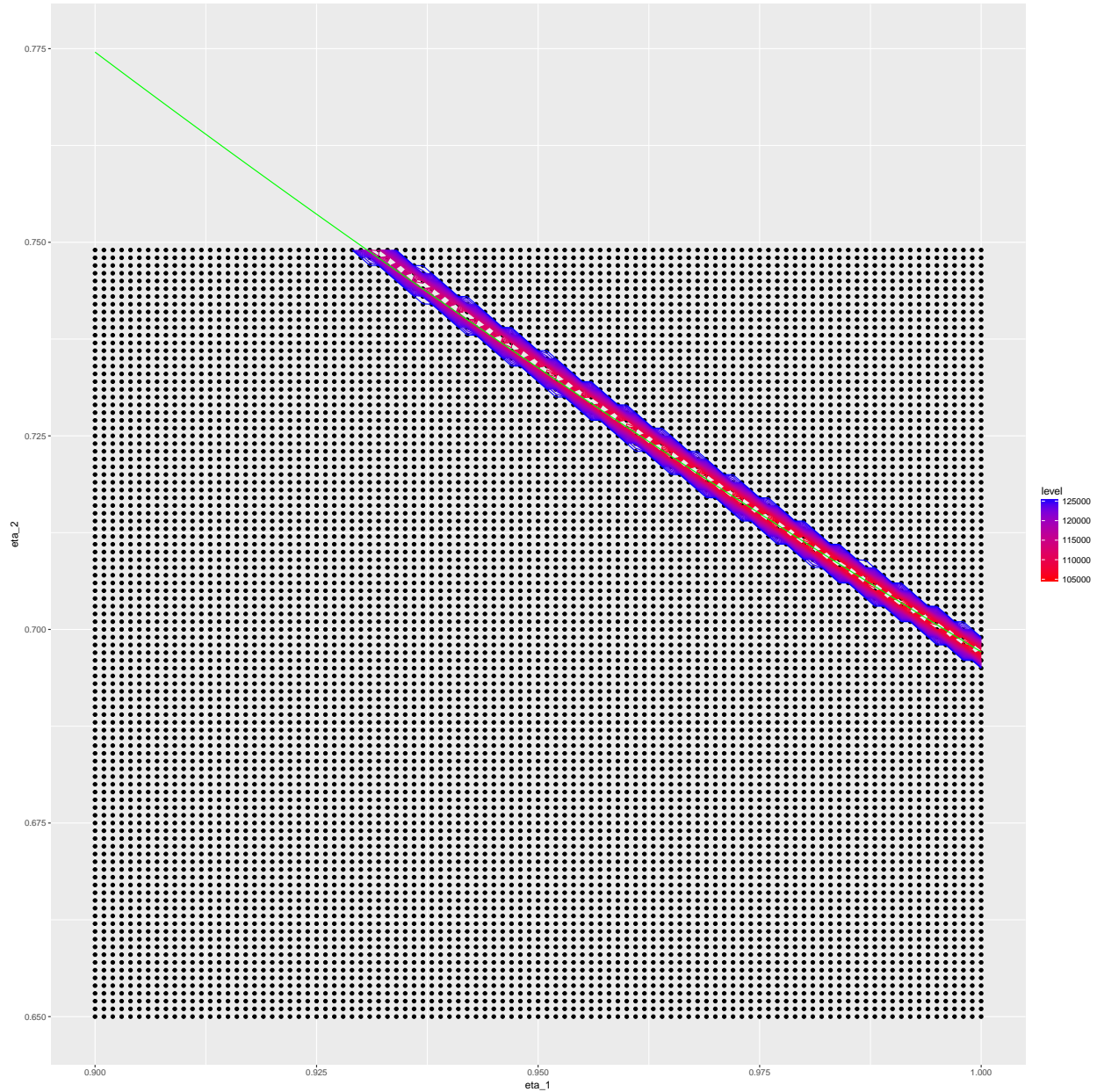
3.1.4 Abschätzung von Wirkungsgraden für die Ladung bzw. die Entladung der Batterie

Durch die Annahme von zwei (konstanten) Wirkungsgraden für den Ladevorgang und den Entladevorgang kann aus den gemessenen Werten für die Batterieladung und -entladung fiktiv ein Ladezustand der Batterie berechnet und mit den Messwerten für den Ladezustand verglichen werden. Durch Auswahl der besten Approximation im Sinn einer zu wählenden Norm ergibt sich eine Schätzung für die beiden Wirkungsgrade.

Gewählte Abstandsnorm: norm_2wurzel_aus_2pot

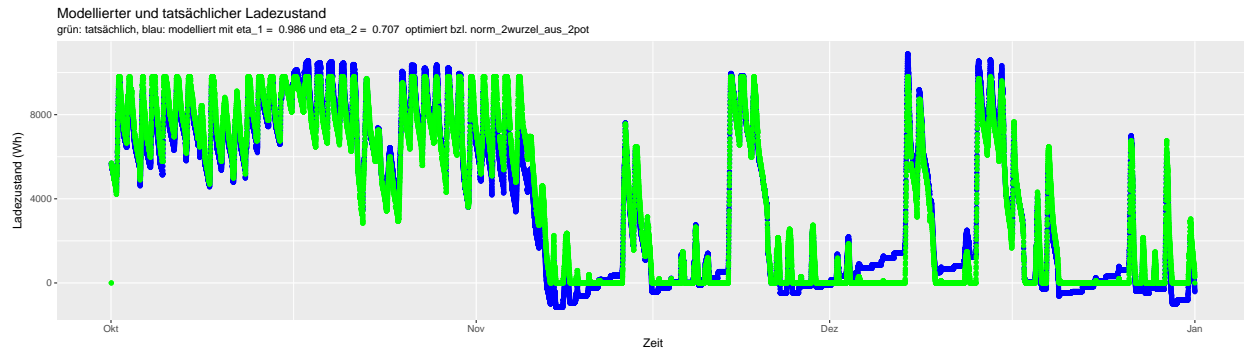
Fuer Ladung und Entladung ergeben sich die Wirkungsgrade 0.986 und 0.707

3.1.5 Visualisierung der Approximationsqualität



Die Modellierung mit Hilfe zweier konstanter Wirkungsgrade ist ersichtlich wenig trennscharf: Die beste Approximationsqualität wird entlang einer Kurve erreicht, die annähernd durch das Produkt η_1 mal $\eta_2 = \text{constant}$ beschrieben wird.

Zum Vergleich: Verlauf des gemessenen und des modellierten Ladezustands



Aus dem Diagramm lässt sich erkennen, dass die Modellvorstellung eines konstanten Wirkungsgrades offensichtlich nur sehr grob gültig ist.

3.2 Strecken monotoner Entladung

Die beiden oben gefundenen Wirkungsgrad für das Laden der Batterie und für den Entladevorgang sollten sich auch wiederfinden lassen, wenn man Perioden beobachtet, in den die Batterie monoton geladen oder entladen wird. Dies ist das Ziel der folgenden Auswertung.

```
#source("04_Auswertungen_Monotone_Entladung_finden.R")
# ZIEL
source("04-Auswertungen_Monotonie_mit_Ent_Ladung.R")
```

```
## Es werden
## 11672 5min-Intervalle negativer und
## 5436 Intervalle positiver Bilanz registriert.
## 9381 Intervalle sind neutral.
```

3.3 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")
```

3.4 Perioden zwischen horizontalen Niveaus bilden - Neutrale Zyklen

Ein *neutraler Zyklus* ist eine Lade-Entlade-Vorgang der von einem Ladezustand des Akkus 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. Ein solcher Zyklus kann jeweils über oder unter dem Ausgangslevel bleiben (später mit UP bzw. DOWN gekennzeichnet).

3.4.1 Bildung der Grundfunktionen

1. Initialisieren der Funktion “zyklus_daten_gen(xdata, l)” mit den Parametern xdata zur Übergabe der Daten und l zur Übergabe des Levels

Erzeugt die Spalten, die einen Zyklus mit einem Zähler charakterisieren und dessen Länge zählen:
zyklus --- len_of_zyklus

2. Initialisieren der Funktion "zyklus_summen_gen(xdata)" mit dem Parameter xdata zur Übergabe von data

Erzeugt die innerhalb eines Zyklus konstanten Werte:

```
max_level : max(ladezustand),  
min_level : min(ladezustand),  
hub_level : max_level - min_level,  
mit_level : (min_level+max_level)/2,  
durchsatz : hub_level/len_zyklus*12, Einheit Wh zwischen Min und Max / Stunde  
signum      : Wenn max_level über dem vorgegebenen Level "UP" sonst "DOWN"  
lev          : Der gewählte Level gespeichert in % im Hinblick auf die Verkettung der Daten zu mehreren
```

3.4.2 Zusammenfassung dieses Vorgangs

Dazu wird folgende Funktion definiert

```
zyklen_bilden <- function(xdata, x) { # xdata =Datensatz, x Vorgabe eines Levels  
  xdata = zyklus_daten_gen(xdata, x)  
  xdata = zyklus_summen_gen(xdata,x)  
  red_data = zyklus_reduzieren(xdata)  
  return(red_data)  
}
```

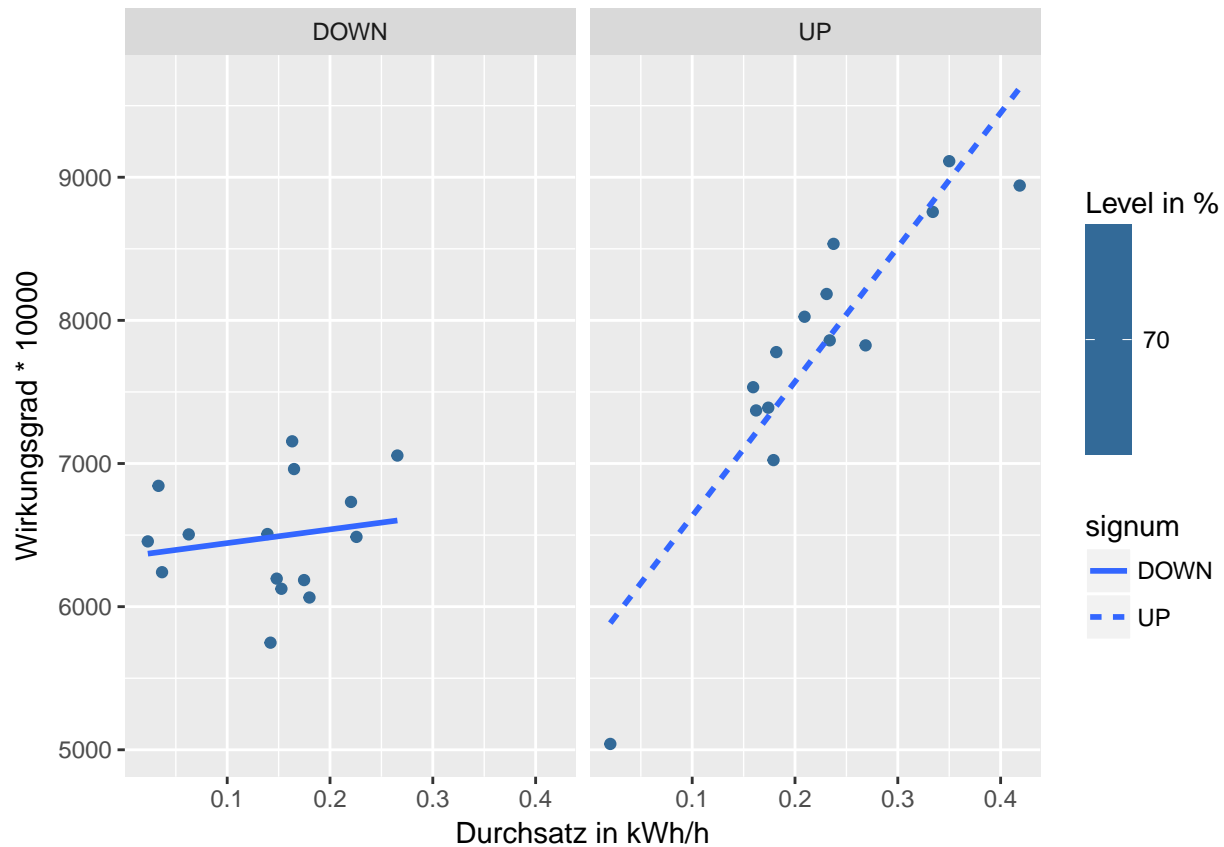
Sie gibt die Auswertung zurück mit jeweils einem Wert pro Zyklus.

4 Graphische Auswertungen

4.1 Darstellung der Wirkungsgrade in Abhängigkeit von Durchsatz

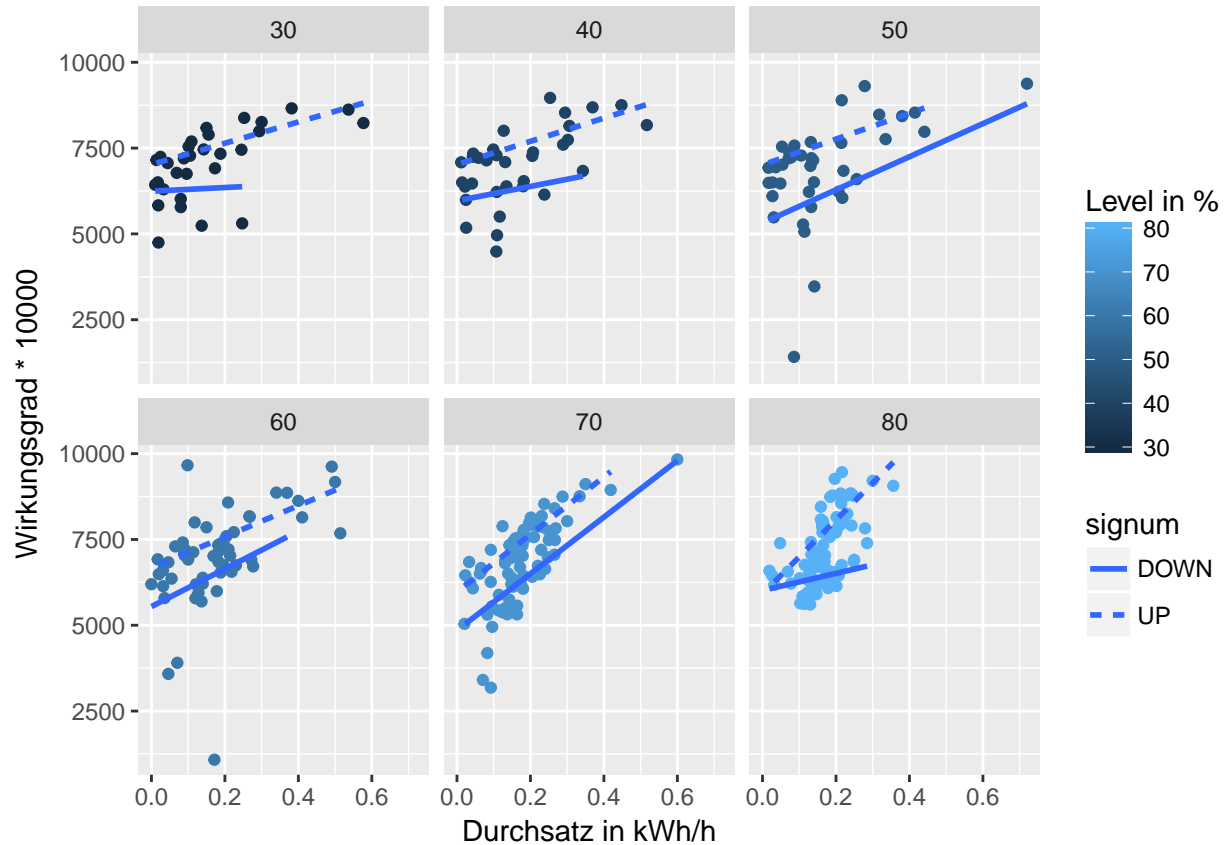
Die maximale Energiedifferenz zwischen höchsten und niedrigsten Wert im Speicher in einer Halbperiode dividiert durch die Dauer der Halbperiode wird als (Energie-)Durchsatz bezeichnet angegeben in Wh/h. 'lev' bezeichnet dem gewählten level dividiert durch 1000.

```
#-----  
# Auswertung der Zyklen  
  
proj_level = zyklen_bilden(data, 7000)  
  
proj_level <- proj_level %>%  
  filter(eta > 0 & eta <= 10000) %>%  
  filter(day >= "2017-11-01")  
  
proj_level %>%  
  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 ),method=lm, se =FALSE) +  
  labs(  
    x = "Durchsatz in kWh/h",  
    y = "Wirkungsgrad * 10000",  
    color = "Level in %" )+  
  facet_wrap(~ signum)
```

Durch Wahl mehrerer Levelwerte erhält man folgende Darstellung

```
some_levels <- c(3000, 4000, 5000, 6000, 7000, 8000)
proj_level <- tibble()
for ( level in some_levels) {
  # zu jedem level den Datensatz auswerten und reduzieren auf eine
  proj_level <- rbind(proj_level, zyklen_bilden(data, level)) # ueber mehrere Level aufsammeln
}
proj_level %>%
  filter(eta <= 10000 & eta != 0) %>%
  filter(day >= "2017-10-01") %>%
  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 ),method=lm, se =FALSE) +
  labs(
    x = "Durchsatz in kWh/h",
    y = "Wirkungsgrad * 10000",
    color = "Level in %"
  ) +
  #+
  #geom_line(mapping = aes(x = durchsatz, y = eta, linetype = signum )) #+
  facet_wrap(~ lev)
```

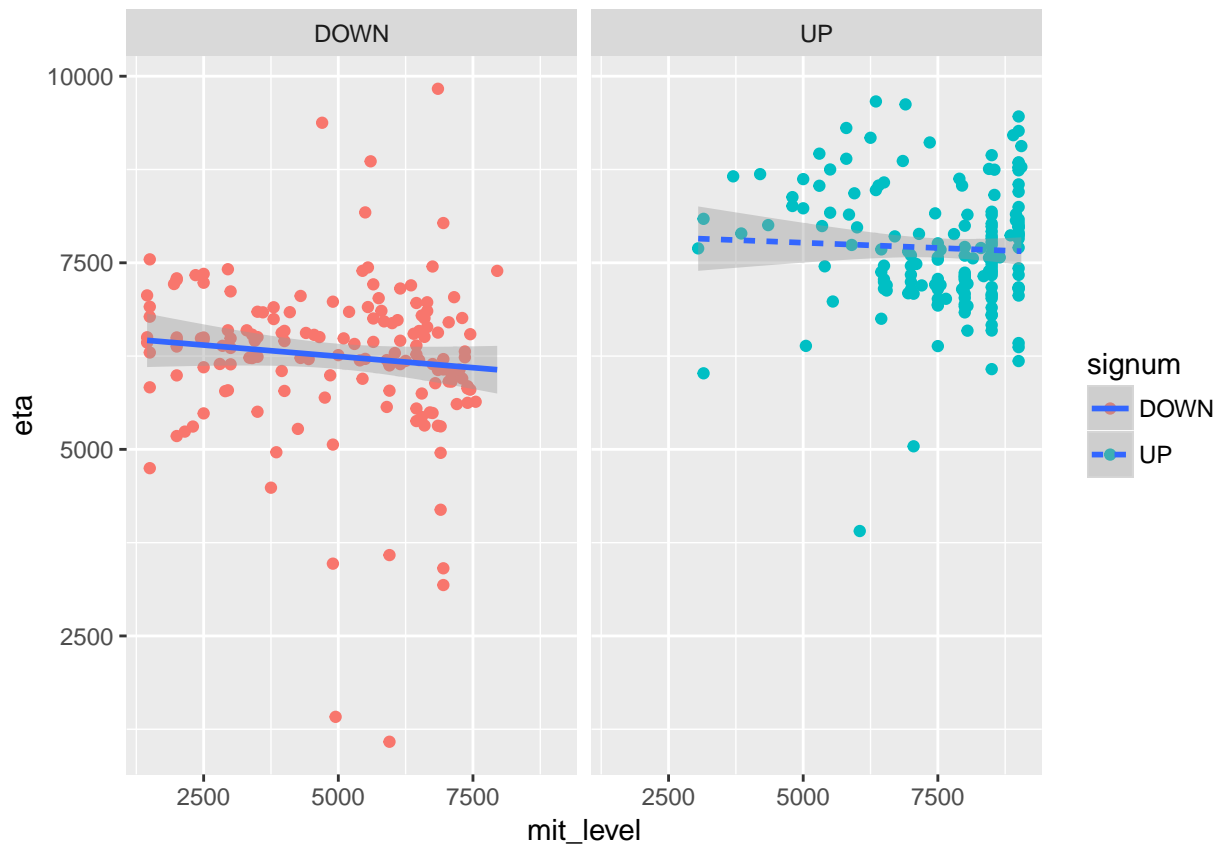


4.2 Darstellung der Wirkungsgrade in Abhängigkeit von der Mitte der Halbperiode

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

```
## level 8000
## # A tibble: 125 x 22
##           zeit batt_ladung batt_entladung ladezustand month week
##           <dtm>         <dbl>         <dbl>         <dbl> <chr> <dbl>
## 1 2017-10-01 00:40:00      0.00000      19.3333333          0    10    40
## 2 2017-10-01 14:10:00    115.08333      0.0000000         8000    10    40
## 3 2017-10-01 14:15:00     87.33333      0.1666667         8100    10    40
## 4 2017-10-02 00:15:00      0.00000      9.0833333         8000    10    40
## 5 2017-10-02 09:40:00    108.50000      0.0000000         8000    10    40
## 6 2017-10-02 09:45:00    163.50000      0.0000000         8200    10    40
## 7 2017-10-02 21:50:00      0.00000     23.5000000         8000    10    40
## 8 2017-10-03 14:30:00    172.08333      0.0000000         8000    10    40
## 9 2017-10-03 14:35:00    105.66667      0.0000000         8200    10    40
## 10 2017-10-03 21:45:00      0.00000     23.9166667         8000    10    40
## # ... with 115 more rows, and 16 more variables: day <date>, hour <dbl>,
## #   ladediff <dbl>, ct <dbl>, zyklus <dbl>, len_zyklus <dbl>,
## #   lev_bat_in <dbl>, lev_bat_out <dbl>, eta <dbl>, max_level <dbl>,
## #   min_level <dbl>, hub_level <dbl>, mit_level <dbl>, durchsatz <dbl>,
```

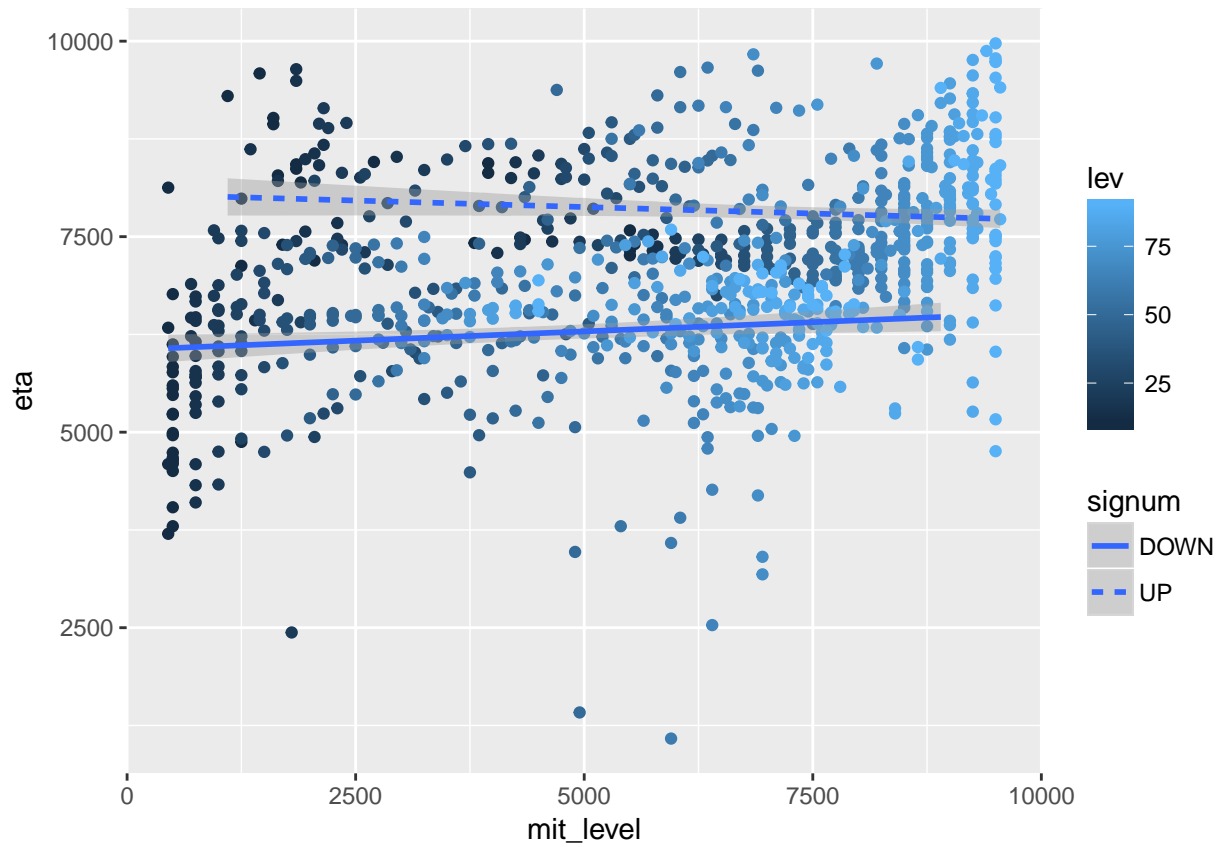
```
## #   signum <chr>, lev <dbl>
```



4.2.1 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) {
  proj_level <- rbind(proj_level, zyklen_bilden(data, level))
}
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 ), method=lm) #+
```



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