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 Größen in der Einheit W, mit Ausnahme von ‘ladezustand’, dieser wird beim Lesen als Prozentsatz übergeben und anschließend auf normiert weil die Batterie eine Kapazität von annähernd besitzt kann dies auch als 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\_01\_12\_2017.csv Daten\_02\_10\_2017.csv Daten\_02\_11\_2017.csv Daten\_03\_10\_2017.csv Daten\_03\_11\_2017.csv Daten\_04\_10\_2017.csv Daten\_04\_11\_2017.csv Daten\_05\_10\_2017.csv Daten\_05\_11\_2017.csv Daten\_06\_10\_2017.csv Daten\_06\_11\_2017.csv Daten\_07\_10\_2017.csv Daten\_07\_11\_2017.csv Daten\_08\_10\_2017.csv Daten\_08\_11\_2017.csv Daten\_09\_10\_2017.csv Daten\_09\_11\_2017.csv Daten\_10\_10\_2017.csv Daten\_10\_11\_2017.csv Daten\_11\_10\_2017.csv Daten\_11\_11\_2017.csv Daten\_12\_10\_2017.csv Daten\_12\_11\_2017.csv Daten\_13\_10\_2017.csv Daten\_13\_11\_2017.csv Daten\_14\_10\_2017.csv Daten\_14\_11\_2017.csv Daten\_15\_10\_2017.csv Daten\_15\_11\_2017.csv Daten\_16\_10\_2017.csv Daten\_16\_11\_2017.csv Daten\_17\_10\_2017.csv Daten\_17\_11\_2017.csv Daten\_18\_10\_2017.csv Daten\_18\_11\_2017.csv Daten\_19\_10\_2017.csv Daten\_19\_11\_2017.csv Daten\_20\_10\_2017.csv Daten\_20\_11\_2017.csv Daten\_21\_10\_2017.csv Daten\_21\_11\_2017.csv Daten\_22\_10\_2017.csv Daten\_22\_11\_2017.csv Daten\_23\_10\_2017.csv Daten\_23\_11\_2017.csv Daten\_24\_10\_2017.csv Daten\_24\_11\_2017.csv Daten\_25\_10\_2017.csv Daten\_25\_11\_2017.csv Daten\_26\_10\_2017.csv Daten\_26\_11\_2017.csv Daten\_27\_10\_2017.csv Daten\_27\_11\_2017.csv Daten\_28\_10\_2017.csv Daten\_28\_11\_2017.csv Daten\_29\_10\_2017.csv Daten\_29\_11\_2017.csv Daten\_30\_10\_2017.csv Daten\_30\_11\_2017.csv Daten\_31\_10\_2017.csv

### Einige Spalten werden erzeugt, gelöscht, bearbeitet und z.B. neu normiert:

1. Die neue Spalte ‘ct’ zaehlt die Datenzeilen
2. Über ‘ladezustand’ laeuft eine Glaettungsfunktion, um einzelne Ausfaelle in den Messungen zu beseitigen.
3. ‘month’, ‘day’ und ‘hour’ werden aus der Variablen ‘zeit’ extrahiert und im Datumsformat “yyyy-mm-dd” bzw. als Zahl 0 - 23 gespeichert.
4. ‘ladediff’ wird als Differenz von Ladezustand zwischen dem aktuellen Zustand und dem vorangegangenen berechnet (Einheit Wh).
5. ‘batt\_ladung’ und ‘bat\_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 im Zeitintervall

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

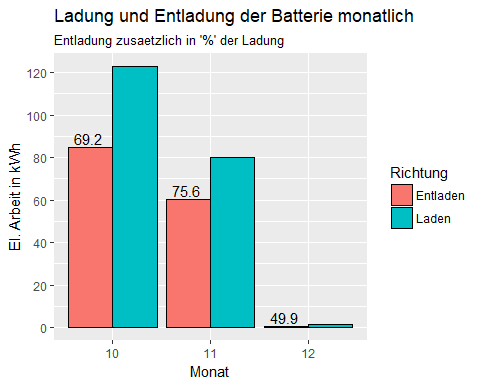
## Erzeuge Tabelle Verbrauch   
## Loesche Spalten aus data leistung.pv leistung.stp netzeinspeisung netzbezug .  
## Der Datensatz enthaelt jetzt 17849 Zeilen.

# Auswertungen

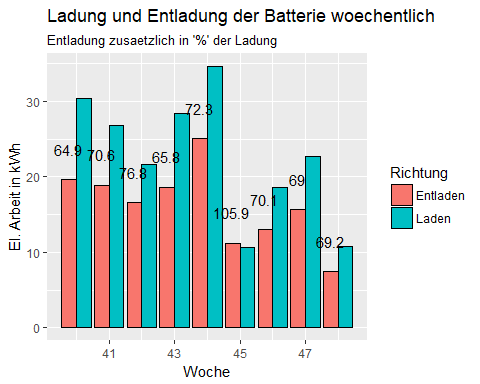
## Einfache Summenbildungen

### Monatssummen

source("04\_Monatliche\_Batterie\_Ladung.R", print.eval=TRUE)



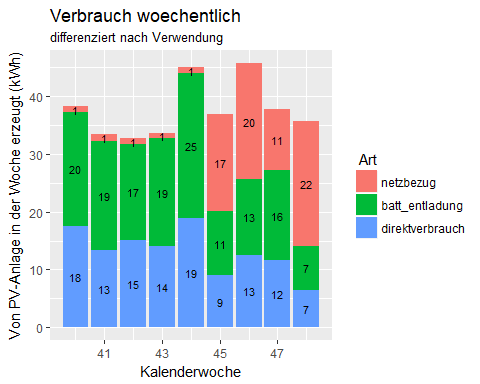
source("04\_Woechentliche\_Batterie\_Ladung.R", print.eval=TRUE)



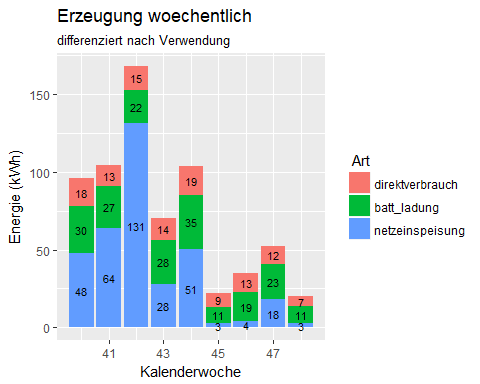
#ggsave("Wochen\_Wirkungsg.pdf")

### Eigenverbrauch - auch fürs Finanzamt

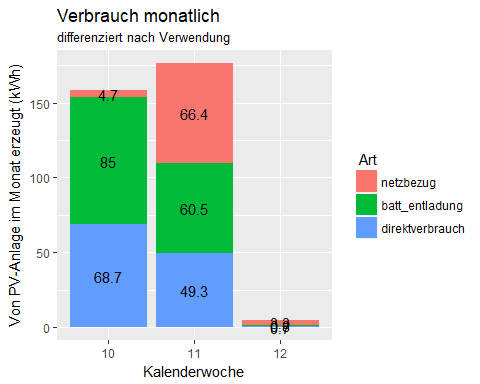
zusfg <- verbrauch %>%   
 group\_by(week) %>%   
 summarise(batt\_ladung = sum(batt\_ladung)/1000,  
 batt\_entladung = sum(batt\_entladung)/1000,  
 leistung.pv = sum(leistung.pv)/1000,  
 netzeinspeisung = sum(netzeinspeisung)/1000,  
 netzbezug = sum(netzbezug)/1000) %>%   
 mutate( direktverbrauch = leistung.pv - netzeinspeisung - batt\_ladung,   
 eigenverbrauch = batt\_entladung + direktverbrauch  
 )  
  
zusfg.long <- melt(zusfg , id = "week", measure = c("netzbezug","batt\_entladung", "direktverbrauch"))  
ggplot(zusfg.long, aes(week, value, fill = variable)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs( x = "Kalenderwoche",  
 y = "Von PV-Anlage in der Woche erzeugt (kWh)",  
 fill = " Art",  
 title = "Verbrauch woechentlich",  
 subtitle = "differenziert nach Verwendung") +  
 geom\_text(aes(label=round(value,-0)), size= 3, data = zusfg.long, position = position\_stack(vjust=0.5))



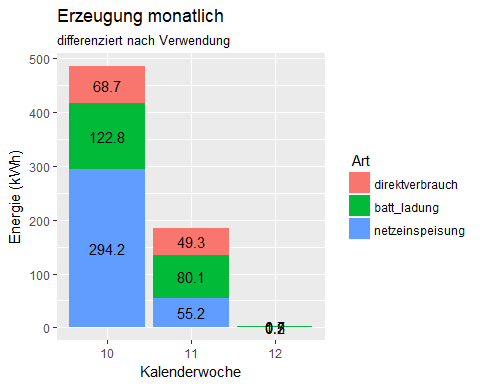
zusfg.long <- melt(zusfg , id = "week", measure = c("direktverbrauch","batt\_ladung","netzeinspeisung" ))  
ggplot(zusfg.long, aes(week, value, fill = variable)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs( x = "Kalenderwoche",  
 y = "Energie (kWh)",  
 fill = " Art",  
 title = "Erzeugung woechentlich",  
 subtitle = "differenziert nach Verwendung") +  
 geom\_text(aes(label=round(value,-0)), size= 3, data = zusfg.long, position = position\_stack(vjust=0.5))



zusfg <- verbrauch %>%   
 group\_by(month) %>%   
 summarise(batt\_ladung = sum(batt\_ladung)/1000,  
 batt\_entladung = sum(batt\_entladung)/1000,  
 leistung.pv = sum(leistung.pv)/1000,  
 netzeinspeisung = sum(netzeinspeisung)/1000,  
 netzbezug = sum(netzbezug)/1000) %>%   
 mutate( direktverbrauch = leistung.pv - netzeinspeisung - batt\_ladung,   
 eigenverbrauch = batt\_entladung + direktverbrauch  
 )  
  
zusfg.long <- melt(zusfg , id = "month", measure = c( "netzbezug", "batt\_entladung", "direktverbrauch"))  
  
ggplot(zusfg.long, aes(month, value, fill = variable)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs( x = "Kalenderwoche",  
 y = "Von PV-Anlage im Monat erzeugt (kWh)",  
 fill = " Art",  
 title = "Verbrauch monatlich",  
 subtitle = "differenziert nach Verwendung") +  
 geom\_text(aes(label=round(value,1)), data = zusfg.long, position = position\_stack(vjust=0.5))

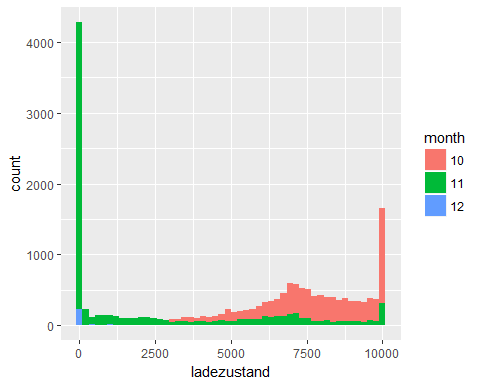


zusfg.long <- melt(zusfg , id = "month", measure = c("direktverbrauch", "batt\_ladung", "netzeinspeisung"))  
  
ggplot(zusfg.long, aes(month, value, fill = variable)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs( x = "Kalenderwoche",  
 y = "Energie (kWh)",  
 fill = " Art",  
 title = "Erzeugung monatlich",  
 subtitle = "differenziert nach Verwendung") +  
 geom\_text(aes(label=round(value,1)), data = zusfg.long, position = position\_stack(vjust=0.5))



### Batteriezustand

batt.zustand <- data %>%   
 select(zeit, month, week, day,hour, ladezustand)   
 ggplot(batt.zustand) +  
 #geom\_bar(mapping = aes(x = ladezustand, fill=month ))   
 stat\_bin(mapping = aes(x = ladezustand, fill=month ), bins = 50)



## 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\_Tagesper.R

## 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).

### 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 charakteriseren und dessen Länge zählen:  
  zyklus --- len\_of\_zyklus

1. 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 Levels

### 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)  
}

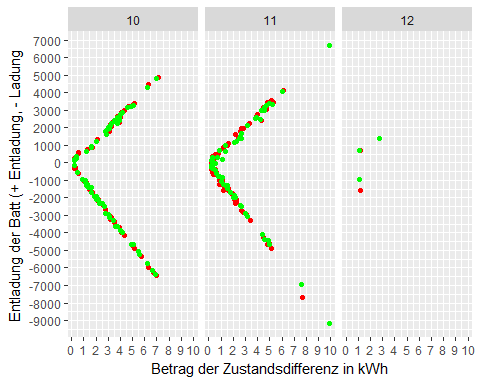
Sie gibt die Auswertung zurück mit jeweils einem Wert pro Zyklus.

## Strecken monotoner Entladung

source("04\_Auswertungen\_Monotone\_Entladung\_finden.R")

temp <- monotonie\_mark(data)  
temp[is.na(temp)] <- 0  
  
temp <- temp %>%   
 mutate( r\_mono = mono,  
 l\_mono = mono)  
  
sp\_vec <- temp %>%   
 filter(mono != 0) %>%   
 mutate(pos = mono\*ct) # pos enthält an Platz p den Wert +/- p wenn ein Anstieg/Abfall vorliegt  
   
sp\_vec = sp\_vec$pos # Vektor der Sprünge  
l\_sp\_vec = length(sp\_vec)-1 # dessen Laenge -1  
  
for (i in 1:l\_sp\_vec) { # Auffüllen nach rechts / links  
 a <- sp\_vec[i] # Sprung bei Pos a  
 o <- sp\_vec[i+1] # nächster bei Betrag von o  
 up <- abs(o)-1 # letzte zu ändernde Position  
 dn <- abs(a)+1 # erste zu ändernde  
 sa <- sign(a)  
 so <- sign(o)  
 for (j in dn:up ) {temp$l\_mono[j] <- so}  
 for (j in dn:up ) {temp$r\_mono[j] <- sa}  
}   
   
temp <- temp %>% select(-one\_of(c("is\_min","is\_max","day\_bat\_in","day\_bat\_out","day\_period\_ladehub","daypd","len\_daypd")))  
  
temp <- temp %>%  
 ungroup() %>%   
 mutate( rup = ifelse(r\_mono == 1, 1, 0) ,   
 rdown = ifelse(r\_mono == -1, 1, 0),  
 rsu = ifelse(lag(rup) != rup, 1, 0),  
 rsd = ifelse(lag(rdown) != rdown, 1, 0),  
 rs = rsu | rsd) %>%   
 mutate( lup = ifelse(l\_mono == 1, 1, 0),  
 ldown = ifelse(l\_mono == -1, 1, 0),  
 lsu = ifelse(lag(lup) != lup, 1, 0),  
 lsd = ifelse(lag(ldown) != ldown, 1, 0),  
 ls = lsu | lsd) %>%   
 ungroup()   
  
temp[is.na(temp)]<- 0  
  
temp <- temp %>%   
 mutate( lnr = cumsum(ls),  
 rnr = cumsum(rs),  
 eins = 1 ) %>%   
 select(-one\_of(c("lsu", "lsd", "ls","rsu", "rsd", "rs"))) %>%   
 group\_by(lnr) %>%   
 mutate(l\_input = sum(batt\_ladung),  
 l\_output = sum(batt\_entladung),  
 l\_netto\_output = (l\_output - l\_input),  
 l\_signum = sign(l\_output-l\_input),  
 l\_entladung= max(ladezustand)-min(ladezustand),  
 l\_dauer = sum(eins) /12 ) %>% # Dauer in Stunden, vorher 1 entspricht 5 min  
 ungroup() %>%   
 group\_by(rnr) %>%   
 mutate(r\_input = sum(batt\_ladung),  
 r\_output = sum(batt\_entladung),  
 r\_netto\_output = (r\_output - r\_input),  
 r\_signum = sign(r\_output-r\_input),  
 r\_entladung= max(ladezustand)-min(ladezustand),  
 r\_dauer = sum(eins) /12 ) %>% # Dauer in Stunden, vorher 1 entspricht 5 min  
 ungroup()   
   
# workleft <- temp %>%   
# group\_by(lnr) %>%   
# slice(1) %>%   
# ungroup()   
#   
# workright <- temp %>%   
# group\_by (rnr) %>%   
# slice(1) %>%   
# ungroup()  
#   
# workleft %>%   
# filter(day >= "2017-10-01") %>%   
# filter(l\_entladung < 10000 & l\_entladung > 100) %>%   
# ggplot(aes(x = l\_entladung, y = l\_netto\_output)) +   
# geom\_point(aes(x = l\_entladung, y = l\_netto\_output, color=l\_dauer)) +  
# geom\_smooth(mapping = aes(x = l\_entladung, y = l\_netto\_output ),method=lm, se =FALSE) +  
# labs(  
# x = "Ladung / Entladung",  
# y = "netto\_output",  
# color = "Dauer" ) +  
# facet\_wrap(~ l\_signum)

temp %>%   
 filter(day >= "2017-10-01") %>%   
 filter(r\_entladung < 9990 & r\_entladung > 100) %>%  
 filter(l\_entladung < 9990 & l\_entladung > 100) %>%  
 ggplot() +   
 #ggplot(aes(x = r\_entladung, y = r\_netto\_output)) +   
 geom\_point(aes(x = l\_entladung/1000, y = l\_netto\_output), color= "red") +  
 geom\_point(aes(x = r\_entladung/1000, y = r\_netto\_output), color= "green") +  
 #geom\_point(aes(x = r\_entladung/1000, y = r\_netto\_output, color= as.logical((1+r\_signum)/2))) +  
 scale\_y\_continuous(breaks = seq(-10000, 10000, by = 1000)) +  
 scale\_x\_continuous(breaks = seq(0, 10, by = 1)) +  
 #geom\_smooth(mapping = aes(x = l\_entladung, y = l\_netto\_output )) + #,method=lm, se =FALSE) +  
 #geom\_smooth(mapping = aes(x = r\_entladung, y = r\_netto\_output ),method=lm, se =FALSE) +  
 labs(  
 x = "Betrag der Zustandsdifferenz in kWh",  
 y = "Entladung der Batt (+ Entladung, - Ladung ",  
 color = "Vorzeichen" ) +  
 facet\_wrap(~ month)

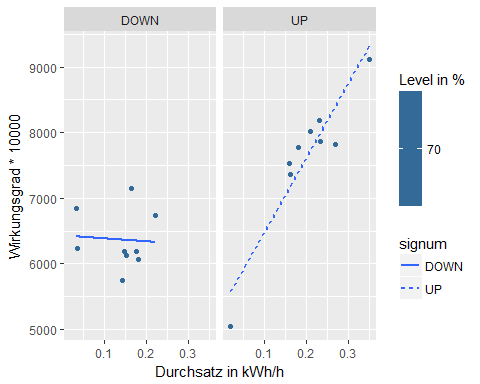


# Graphische Auswertungen

## Darstellung der Wirkungsgrade in Abhängigkeit von Durchsatz

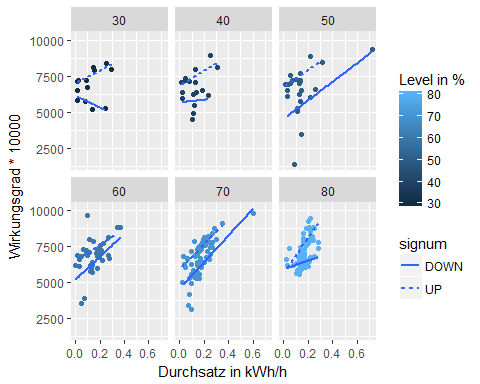
Die maximale Energiedifferenz zwischen höchsten und niedrigsten Wert im Speicher in einer Halbperiode dividiert durch die Dauer der Halbperiode wird als (Energie-)Durchsatz bezeichnet angegeben in Wh/h. ‘lev’ bezeichnet dem gewählten level dividiert durch 1000.

#--------------------------------------------------------   
# 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)



Duch Wahl mehrerer Levelwerte erhält man folgende Darstellung

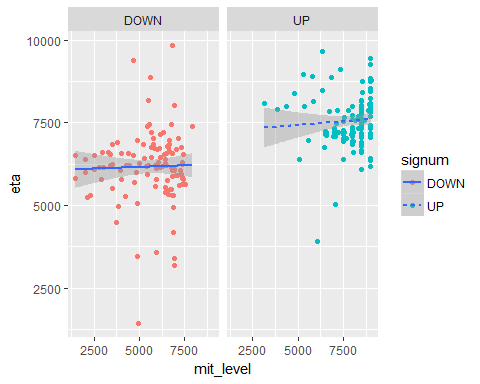
some\_levels <- c(3000, 4000, 5000, 6000, 7000, 8000)  
proj\_level <- tibble()  
for ( level in some\_levels) { # zu jedem level den Datensatz auswerten und reduzieren auf einen Wert pro Periode  
 proj\_level <- rbind(proj\_level, zyklen\_bilden(data, level)) # ueber mehrere Level aufsammeln  
}  
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)



## 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)

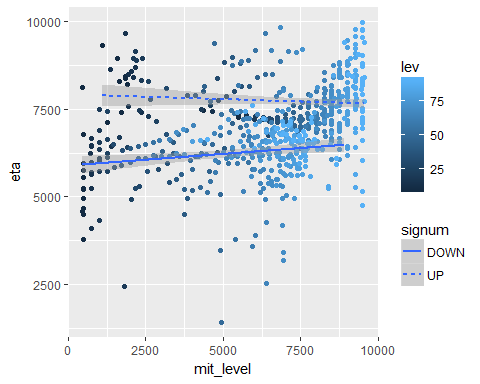
## level 8000   
## # A tibble: 109 x 29  
## zeit batt\_ladung batt\_entladung ladezustand month week  
## <dttm> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 2017-10-01 00:40:00 0.00000 19.3333333 0 10 40  
## 2 2017-10-01 14:10:00 115.08333 0.0000000 8000 10 40  
## 3 2017-10-01 14:15:00 87.33333 0.1666667 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 99 more rows, and 23 more variables: day <date>, hour <dbl>,  
## # ladediff <dbl>, ct <dbl>, is\_min <dbl>, is\_max <dbl>, daypd <dbl>,  
## # len\_daypd <dbl>, day\_bat\_in <dbl>, day\_bat\_out <dbl>,  
## # day\_period\_ladehub <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>



### Das Gleiche mit Aufsammeln 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, 8500, 9000)  
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)