

# tübix

## Datenanalyse einer Solaranlage mit python

Dr. Olaf Flebbe  
of ät oflebbe.de

tübix

24.6.2017 Tübingen

# About me

PhD in computational physics  
(Theor. Astrophysik Tübingen)

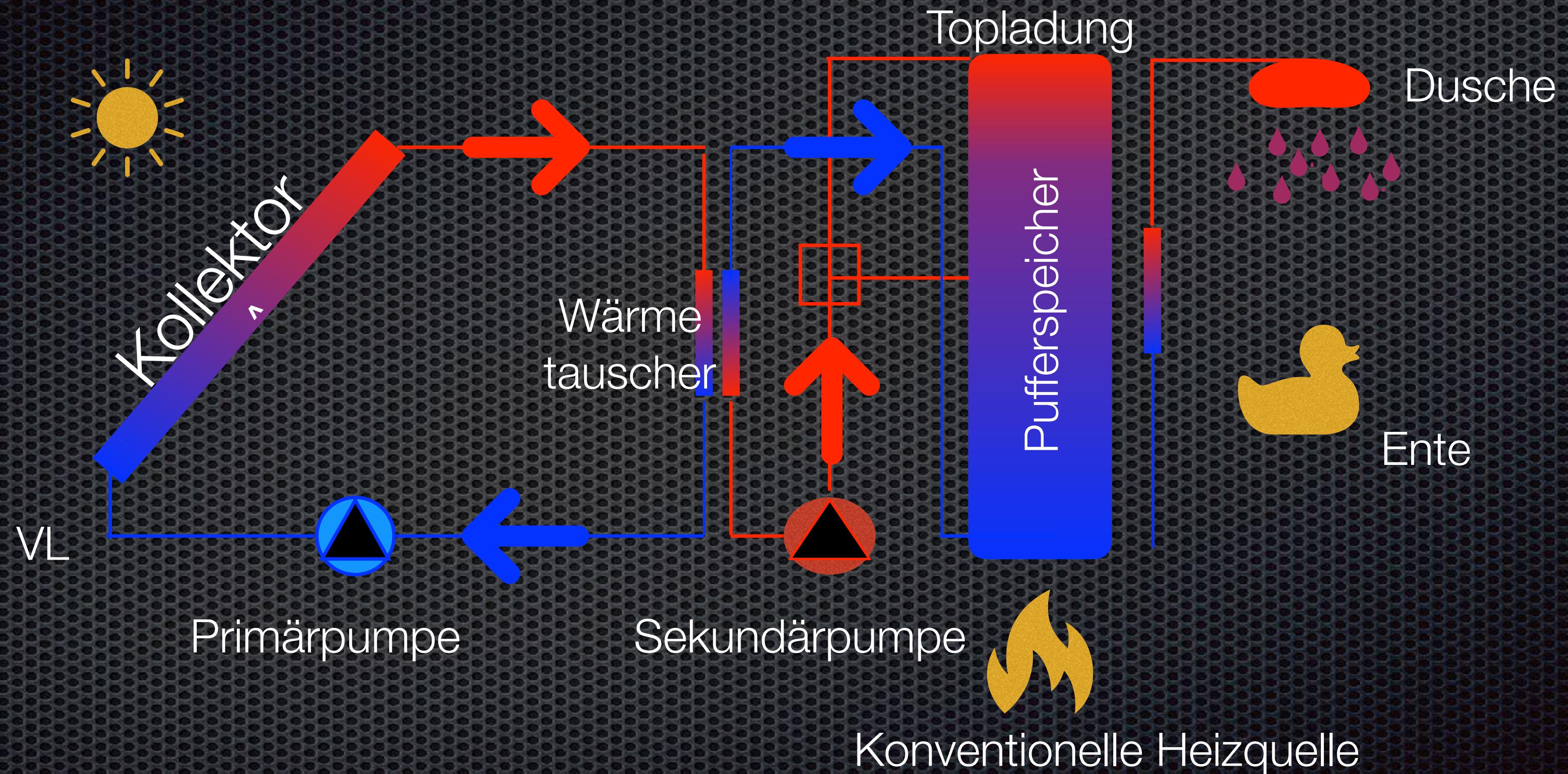
Former projects: Minix68k (68k FP Emulation), Linux libm.so.5 (High Precision FP), perl and python for epoc, flightgear, msktutil

Member Apache Software Foundation, PMC Apache Bigtop

Software Architect  
Robert Bosch GmbH



# Logging der th. Solaranlage



Sie befinden sich hier: Produkte / Frei programmierbar / UVR1611

# UVR1611

## Frei programmierbare Universalregelung



UVR1611K-N mit Konsole für  
Wandmontage



### Neue Konsole, neue Einbauversion, neu mit 11 Relais!

Der Universalregler UVR1611 kann durch Funktionsmodule frei programmierbar auf jede Anlagenkonfiguration für das Heizraummanagement abgestimmt werden. Mit dem Interface C.M.I. ist es möglich, die Steuerung über LAN oder Webportal zu bedienen, zu überwachen und am PC, Smartphone oder Tablet eine interaktive Visualisierung dazustellen. Mit dem CAN-Monitor (CAN-MT) bzw. CAN-Touch ist eine Fernbedienung des Reglers möglich. Weitere CAN-Buskomponenten ermöglichen es, die Zahl der Ein- und Ausgänge zu vergrößern. Die Programmerstellung erfolgt mit der Software TAPPS2.

Achtung: Nachfolger für dieses Produkt ist der [UVR16x2 Universalregler](#)



Eingänge: 16



Ausgänge: 11+2



Schnittstellen: DL-Bus, CAN-Bus

# Datenlogger

- D-LOGG
- Problemanalyse
- WinSol
- <http://d-logg-linux.roemix.de>

## Datenkonverter



# Crosscompilation

- Download von Synology Open Source Project
  - <https://sourceforge.net/projects/dsgpl/>
- Download DS214 -> ArmadaXP
- Installieren und verwenden (Debian8 64Bit)

```
apt-get update
apt-get upgrade
apt-get install wget unzip xz-utils
cd /usr/local/
apt-get install xz-utils
tar xf /tmp/armadaxp-gcc493_glibc220_hard-GPL.txz
wget http://d-logg-linux.roemix.de/files/dlogg-linux.zip
unzip dlogg-linux.zip
cd dlogg-linux/dl-aktuelleDaten/
/usr/local/arm-unknown-linux-gnueabi/bin/arm-unknown-linux-gnueabi-gcc \
    dl-aktuelle-datenx-no-curses.c
```

# Install

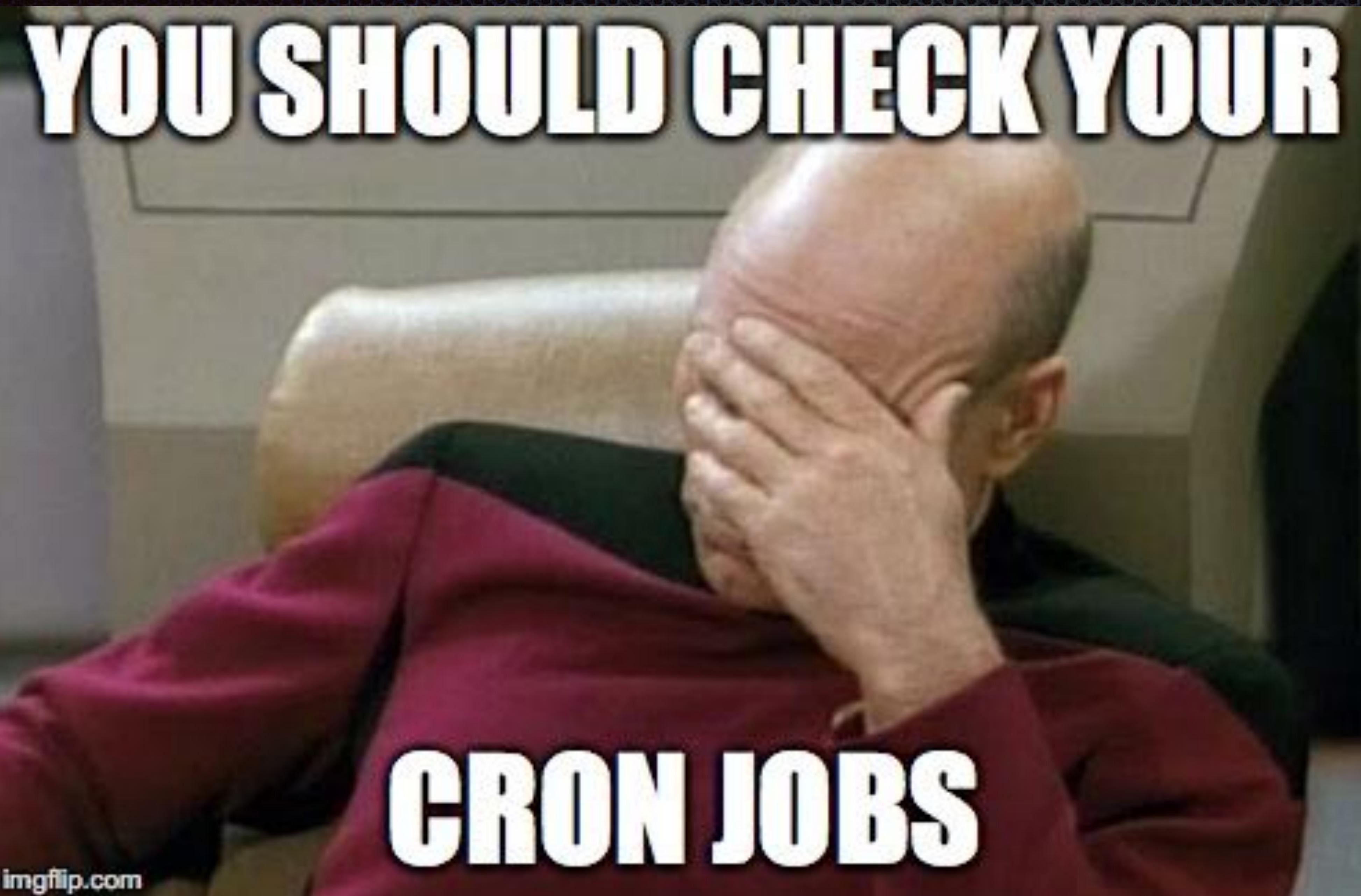
- Das Programm läuft einfach so, aber die USB serielle Schnittstelle muss noch aktiviert werden.
- Alle 30 Sek. ein Datenpunkt
- Cronjob used:

```
#!/bin/sh
if [ ! x`ps | grep dl-aktuelle-daten-no-curses| grep -v grep` = x ] ; then
    exit 0;
fi
insmod /lib/modules/usbserial.ko
insmod /lib/modules/ftdi_sio.ko
cd /volume1/solar
y=`date +%Y` 
if [ ! -d $y ] ; then mkdir $y; fi
cd $y
./dl-aktuelle-daten-no-curses -p /dev/ttyUSB0 --csv >>LOG 2>&1 &
```

# Ausgabe der Datei

```
Datum ; Zeit ; Sens1 ; Sens2 ; Sens3 ; Sens4 ; Sens5 ; Sens6 ; Sens7 ; Sens8 ; Sens9 ; Sens10 ; Sens11 ; Sens12; Sens13 ; Sen  
s14 ; Sens15 ; Sens16 ; Ausg1 ; Drehzst_A1 ; Ausg2 ; Drehzst_A2 ; Ausg3 ; Ausg4 ; Ausg5 ; Ausg6 ; Drehzst_A6 ; Ausg7 ; Drehzst  
_A7 ; Ausg8 ; Ausg9 ; Ausg10 ; Ausg11 ; Ausg12 ; Ausg13 ; kW1 ; kWh1 ; kW2 ; kWh2 ;  
Sens1 ; Sens2 ; Sens3 ; Sens4 ; Sens5 ; Sens6 ; Sens7 ; Sens8 ; Sens9 ; Sens10 ; Sens11 ; Sens12; Sens13 ; Sens14 ; Sens15 ; S  
ens16 ; Ausg1 ; Drehzst_A1 ; Ausg2 ; Drehzst_A2 ; Ausg3 ; Ausg4 ; Ausg5 ; Ausg6 ; Drehzst_A6 ; Ausg7 ; Drehzst_A7 ; Ausg8 ; Au  
sg9 ; Ausg10 ; Ausg11 ; Ausg12 ; Ausg13 ; kW1 ; kWh1 ; kW2 ; kWh2 ;  
  
15.06.17;15:40:48; 107.7; 85.4; 74.3; ---; 73.4; 77.8; 112.6; ---; ---; ---; ---; ---; ---; ---; ---; 44.0; 1;13; 0; 0; 1  
; 1; 0; 0; 0; 0; 0; 1; 0; 0; 0; 2.0;39102.7; ---; ---; ;  
15.06.17;15:41:18; 107.3; 85.3; 74.3; ---; 73.6; 78.1; 112.7; ---; ---; ---; ---; ---; ---; 44.0; 1;14; 0; 0; 1  
; 1; 0; 0; 0; 0; 0; 1; 0; 0; 0; 1.9;39102.7; ---; ---;
```

# YOU SHOULD CHECK YOUR

A close-up photograph of a bald man with a mustache, wearing a Starfleet officer's uniform consisting of a maroon shirt over a green undershirt. He is sitting in a light-colored armchair, holding his hands up to his face in a gesture of distress or despair. The background is dark and out of focus.

# CRON JOBS

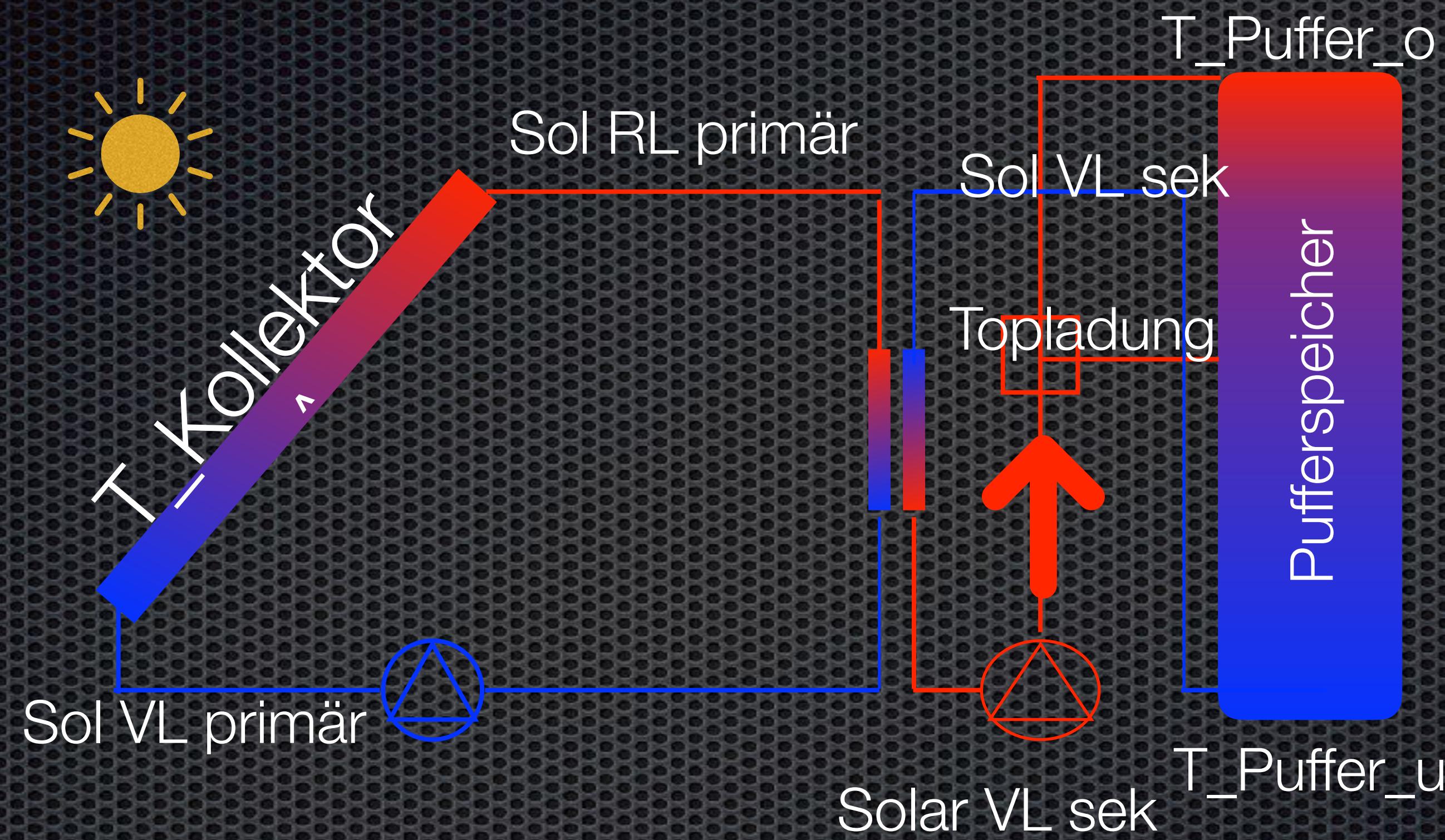
# Install

## ► Cronjob Fehler

```
#!/bin/sh
if [ ! x`ps aux | grep dl-aktuelle-daten-no-curses| grep -v grep` = x ] ; then
  exit 0;
fi
insmod /lib/modules/usbserial.ko
insmod /lib/modules/ftdi_sio.ko
cd /volume1/solar
y=`date +%Y` 
if [ ! -d $y ] ; then mkdir $y; fi
cd $y
../dl-aktuelle-daten-no-curses -p /dev/ttyUSB0 --csv >>LOG 2>&1 &
```

D.h. Jede Stunde wurde ein Programm gestartet, das sich um die serielle Schnittstelle streitet und potentiell den Output kaputtmacht ... !!!

# Logging der Th. Solaranlage



# Vorverarbeitung

- Datenlogging lief
- Die Leerzeichen in der ersten Zeile stören, zweite und dritte Zeile können weg.
- Am einfachsten mit

```
for i in data/solar/201*/*csv; do  
    gsed -i -e 2,3d -e '1s/ //g' $i  
done
```

(Note gsed on MacOS installed from HomeBrew, sed on Linux)

# Verwendete Python Frameworks

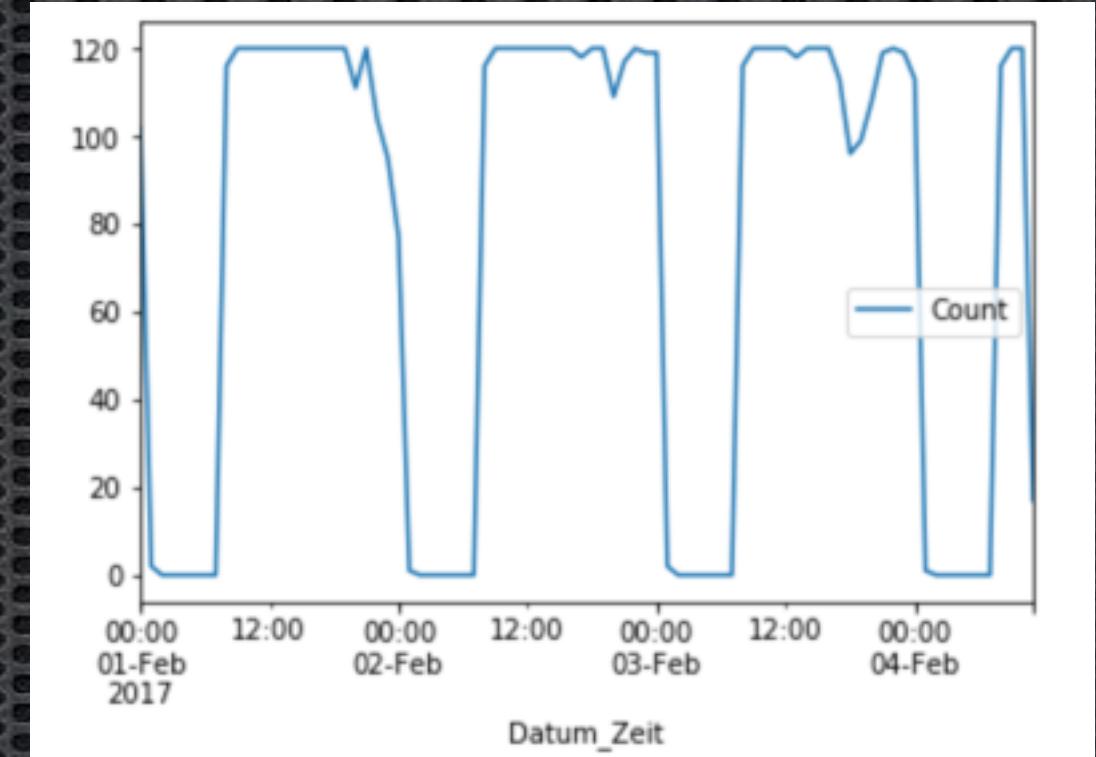
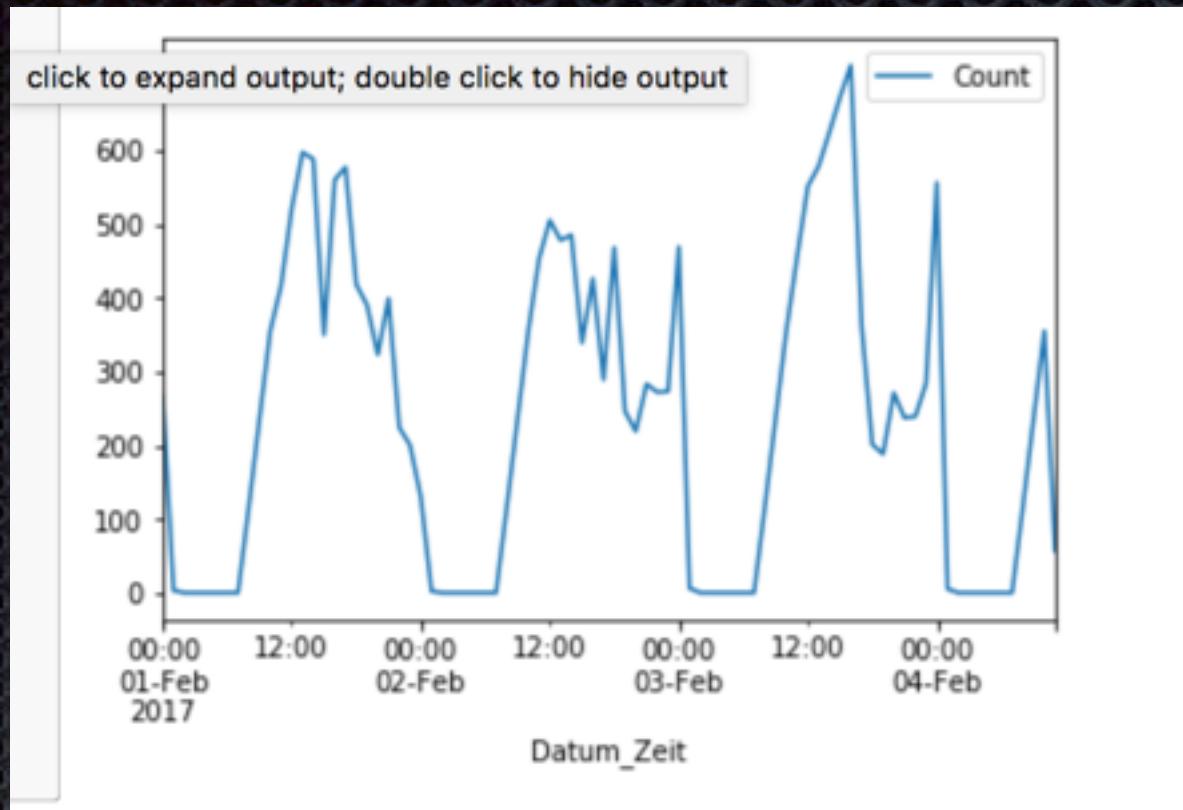
- python Programmiersprache
- Pandas: Datenauswertung
- Jupyter Notebook: Auswertung im Web
- Matplotlib: Plot Bibliothek
- Numpy numerical python
- Bokeh uvam

# Python installieren

- Linux Distro meist unvollständig oder veraltet, module mit pip nachinstallieren.
- MacOSX: Homebrew tut ganz gut, einzelne Module mit pip nachinstallieren
- Windows: Mühselig
- Allen gemeinsam: schlechte Unterstützung von HTTP Proxies.
- Lösung: Anaconda von Continuum Analytics:  
<https://continuum.io>
- Hint: Installiert python3.6 version wenn ihr neu beginnt.

# Beispiel Solar-readin

- Starten des Notebooks mit "jupyter notebook"
- Python Code + Plots im Browser
- Beispiel : solar-readin
- pandas read\_csv ist extrem mächtig
- pandas verwendet dataframes: Das sind Tabellen deren Zeilen durch einen Index adressiert werden aber auch Spaltenweise genutzt werden können.
- Der Index kann eine Zeit sein "Zeitreihen"
- Plotten von Daten mit matplotlib ist eingebaut
- fehlerhaften cronjob von oben mit resampling fixen.



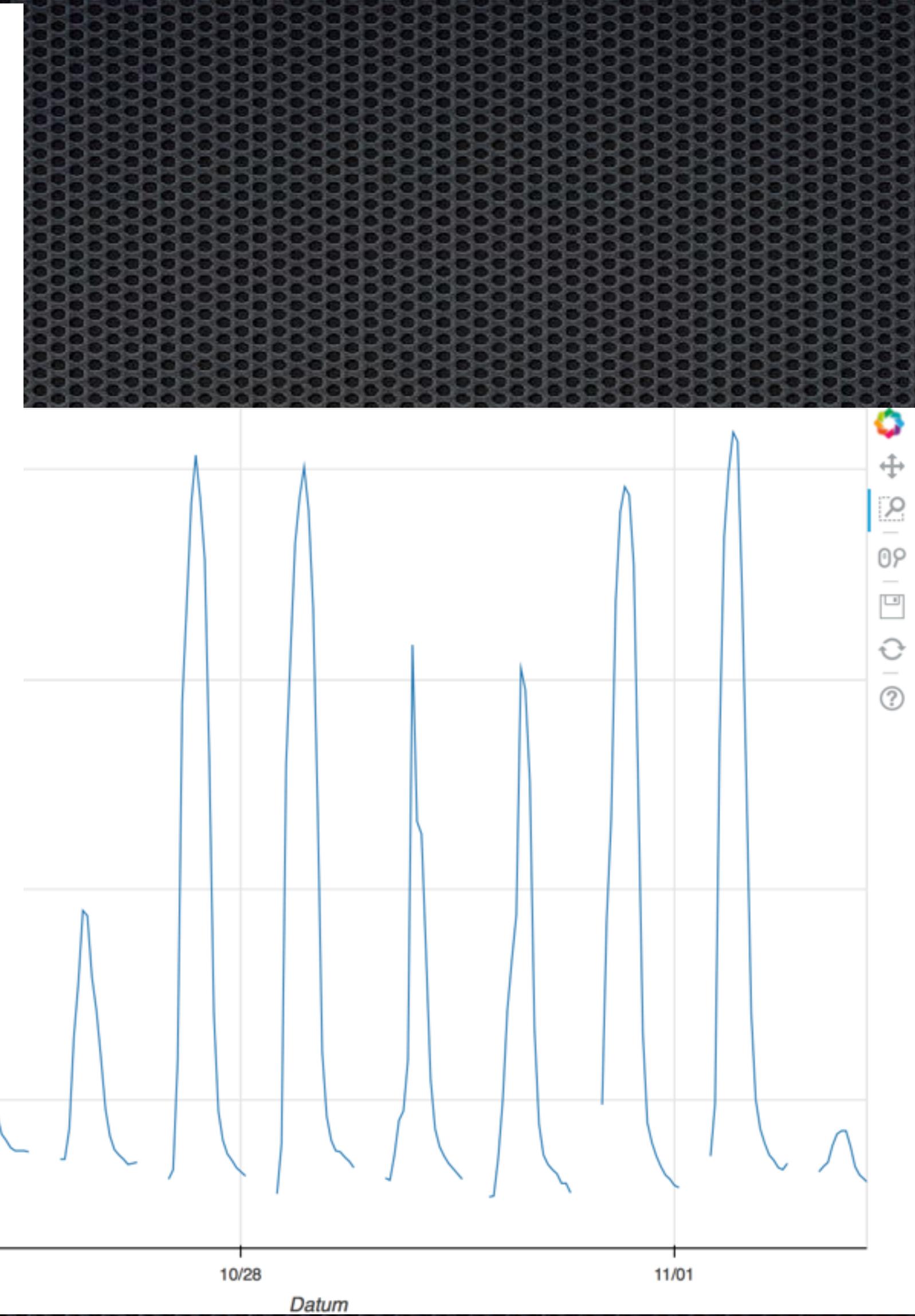
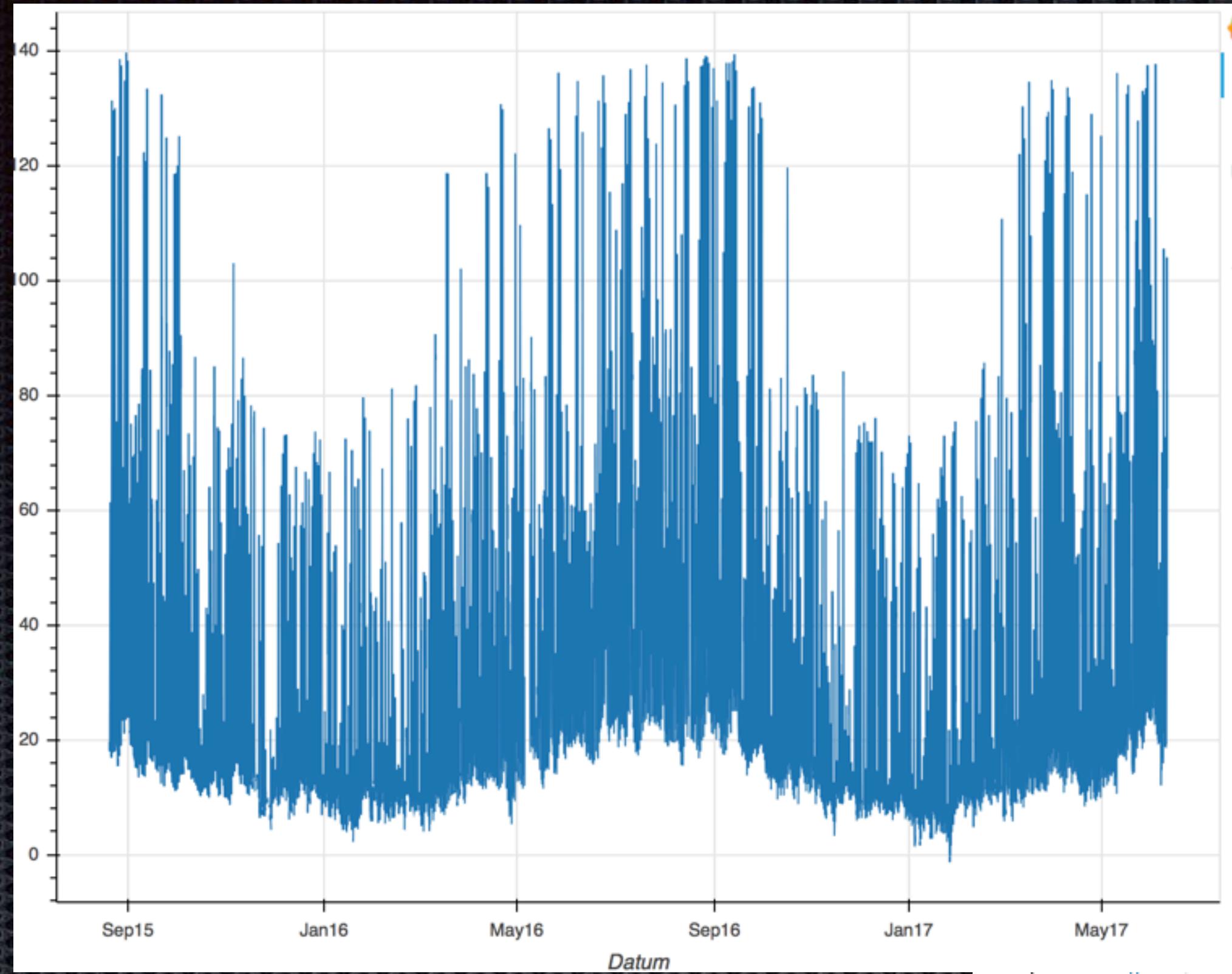
Anzahl der Datenpunkte pro Stunde,  
wg. Fehler in Crontab  
Rohdaten vs. Gefilterte Daten

# Import

- python multiprocessing passt mit pandas zusammen
- CSV Format sehr aufwändig
- HDF5 kompakt und sehr schnell (pytables)  
1.5 GB vs 300MB  
Minuten vs Sekunden
- convert.py läuft mehrere Minuten

# Bokeh

- Interaktive Grafiken
- Dokumentation gewöhnungsbedürftig
- Highlevel API mit aktueller Release entfernt, nun wird weitere Bibliothek (Holoviews) on top empfohlen
- Beispiel: auswertung-bokeh
- (plotly als Konkurrent)



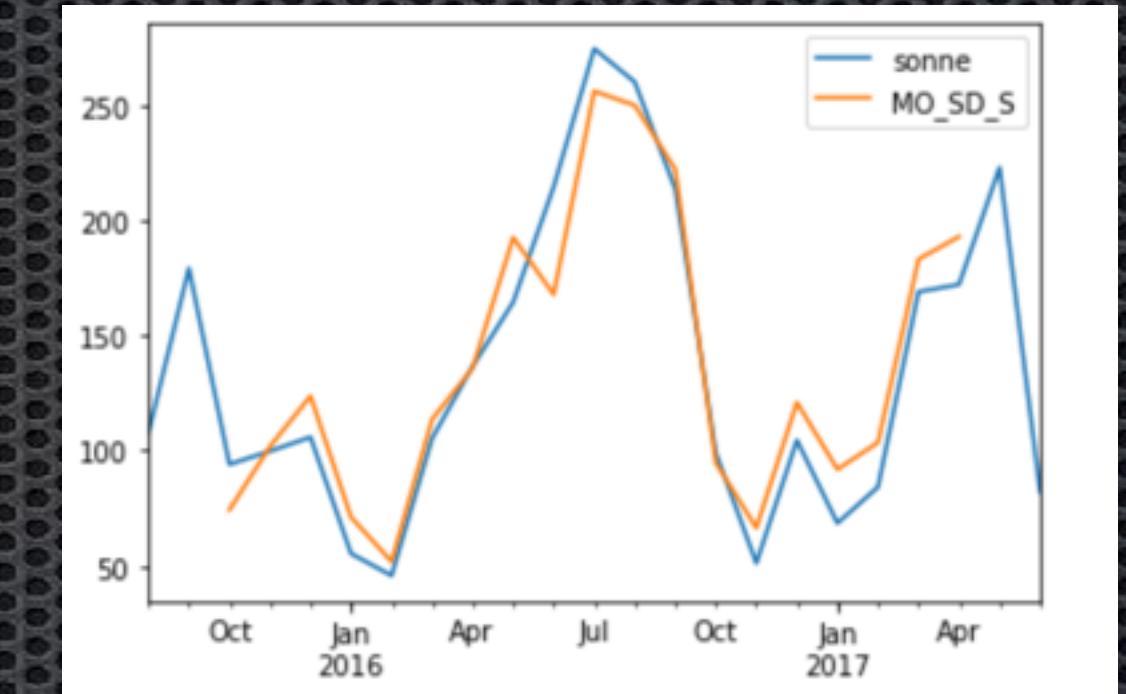
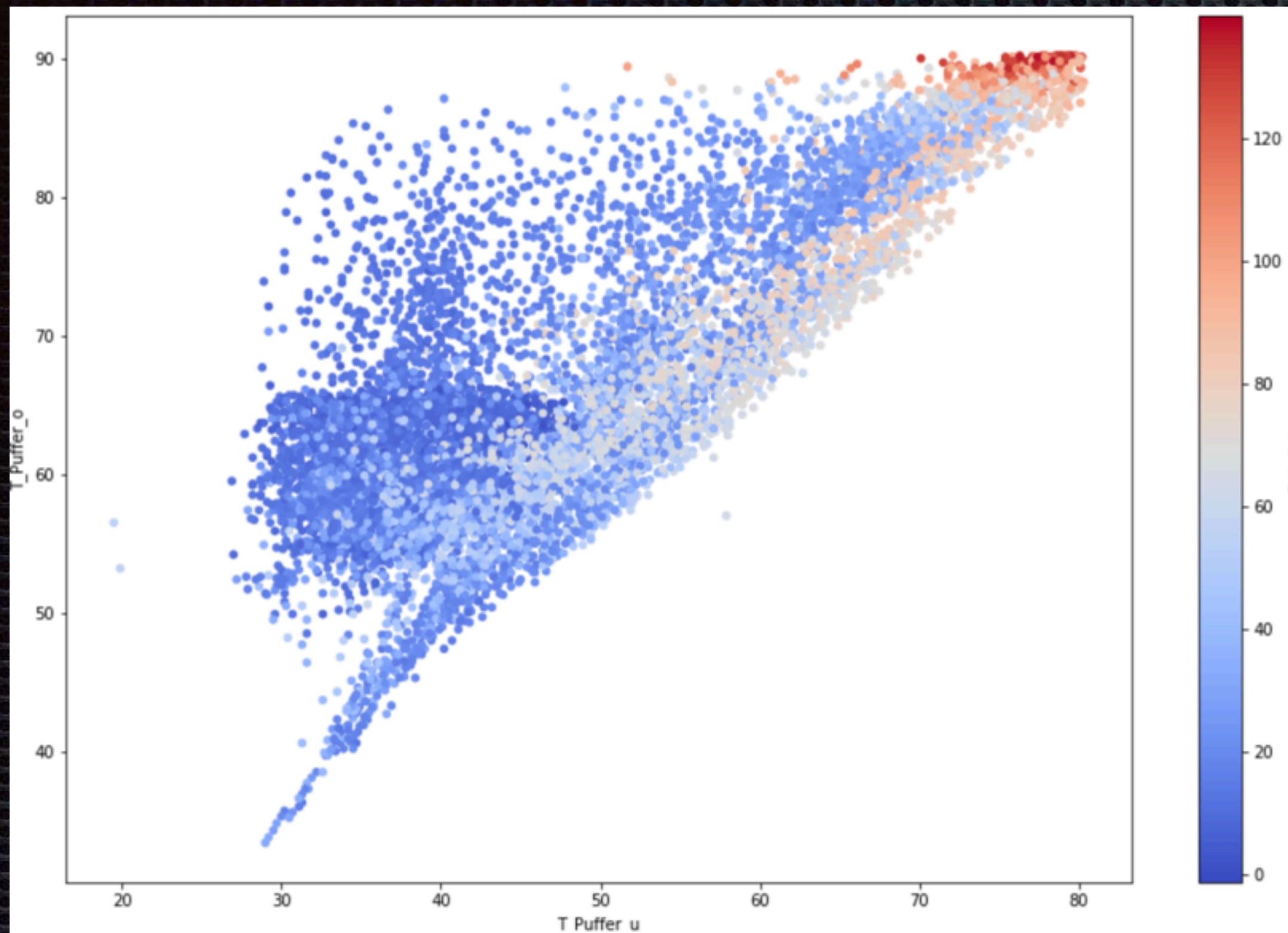
Interaktives Zoom mit Bokeh

# Auswertung Inline

- solar\_auswertung\_inline
- Plot
- Scatterplots
- in pandas eingebaut

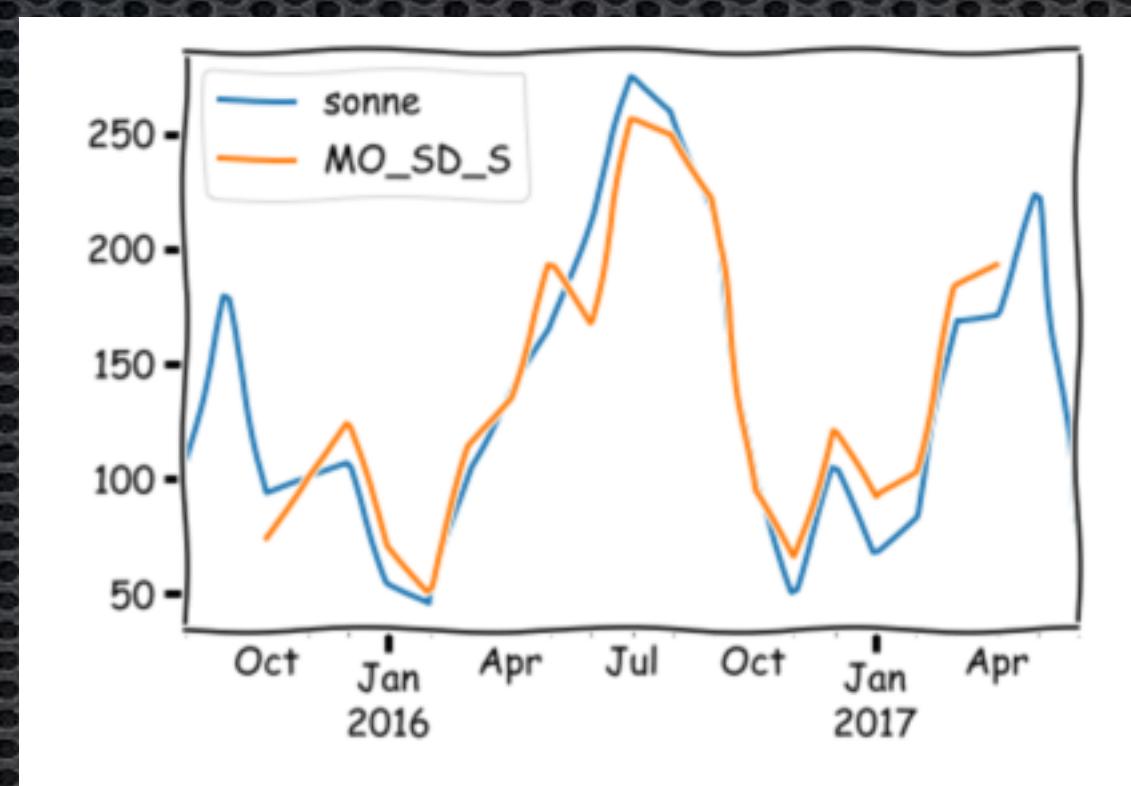
# Auswertung Zeit

- solar\_auswertung\_tz
- Zeitzonen umrechnung (Sommer/Winterzeit)
- Aggregation an Uhrzeiten
- Verschiedene Daten an gemeinsamen Index ausrichten
- Vergleich T\_Kollektor > 45 °C mit DWD Daten  
Sonnenscheindauer



Sonnenscheindauer vs  
 $T_{\text{Kollektor}} > 45$

Scatterplots Temperaturprofil



Nerd Features included

# Lessons Learned März-Juni

- pandas von 0.19.2 nach 0.20.2
  - Stabil, aber gruftige Funktionalitäten entfernt
  - Stackoverflow hilfreich, aber oft gruftige Zugänge
- bokeh 0.12.4 - 0.12.6
  - Documentation viel besser
  - API mehrmals gebrochen, Funktionalitäten entfernt
- Selbst der DWD ändert sein Datenformat zum 1.6.

# Datenanalyse

- "Citizen Science" macht Spass
- Unglaublich mächtige Werkzeuge sind Open Source
- Rohdaten sind wertvoll
- Die Daten und Skripte sind erhältlich von  
<http://www.oflebbe.de/solar-notebook.tgz>

# Nicht behandelt

- dask: Parallelisierung
- scikit: Machine Learning
- Apache Spark: Parallelisierung, SQL auf Daten für ETL
- Apache Zeppelin: Spark Notebook



Noch Fragen?  
of ät oflebbe.de