

# StudyDB

Maria Müller, Michael Sué  
& Stefan Vollmar

vollmar@sf.mpg.de



Max Planck Institute  
for Metabolism Research

for Metabolism Research



# Database Projects (1)



DV-Treffen Göttingen 2019, »ScoreDB: Django-Anwendung im Labor mit Tablets und Barcode-Scanner«,

<https://wiki.init.mpg.de/share/Workshops/DVTreffen/36DVT?action=AttachFile&do=view&target=dv-treffen-2019-scoredb-v08p.pdf>

DV-Treffen Online 2021, »Django-Anwendungen mit Archivierung«,

<https://wiki.init.mpg.de/share/Workshops/DVTreffen/38DVT?action=AttachFile&do=view&target=vortrag-dv-vollmar-2021-v16p.pdf>

# Database Projects (2)



# StudyDB

arten Sie in Ihrem Leben tetenommen  
Sie für jede Spielart eine Antwort: «ni  
einmal pro Woche» oder «ein bis mehr

	niemals	weniger als einmal pro Woche	ein bis mehr- mals pro Woche	
a.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Karten spiele
b.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Pferdewetter (vor Ort, über
c.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sportwetten
d.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Würfelspiel
e.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Spiele im K
f.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Lotterie- od
g.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Bingo um C



DV-Treffen Online 2020, »Django-Datenbanken im Labor, Beispiel GeneDB«,  
<https://wiki.init.mpg.de/share/Workshops/DVTreffen/37DVT?action=AttachFile&do=view&target=vollmar-dv-treffen-2020-genedb-v14.pdf>

DV-Treffen Online 2021, »Django-Anwendungen mit Archivierung«,  
<https://wiki.init.mpg.de/share/Workshops/DVTreffen/38DVT?action=AttachFile&do=view&target=vortrag-dv-vollmar-2021-v16p.pdf>

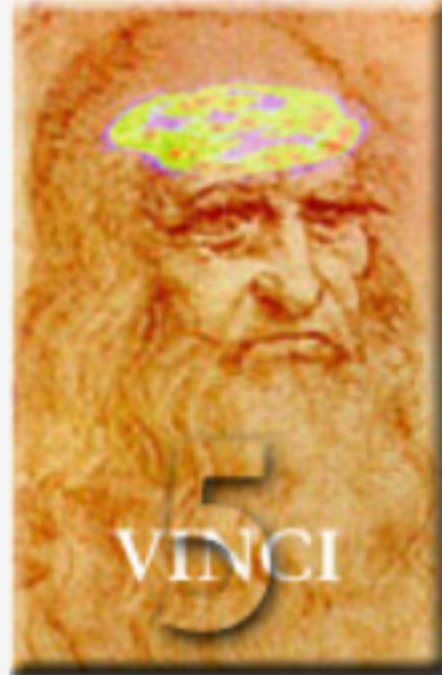
TVADB  
MPI-SF



TVADB  
Uni Köln



# Other Projects



**V**olume  
**I**maging in  
**N**eurological Research  
**C**o-Registration and ROIs  
**I**ncluded

- Visualisation and Analysis of Tomographical Data
- users in major imaging labs originally due to a collaboration with CTI/Siemens (licensed for the HRRT PET scanner)
- C++, Qt, Linux/Mac/Win (we even had a decent Solaris version of VINCI 3.x when Solaris 10 was young; stone tablets have been found attesting to VINCI versions being used **as early as 2003**)
- <https://vinci.sf.mpg.de>

# »Single Source of Truth« (1)

## JSON:

(but would work in principle with **XML** or **YAML**, too)

challenge for StudyDB: get clinicians to (really) describe all required metadata (help needed)

```
{
  "study": "btx",
  "model": "today_medication",
  "fields": [
    {
      "name": "pat_id",
      "type": "pat_id"
    },
    {
      "name": "visite",
      "type": "integer",
      "min": 0,
      "max": 5
    },
    {
      "name": "betablocker",
      "type": "enum",
      "values": [
        "bisoprolol",
        "carvedilol",
        "metoprolol",
        "propranolol",
        "andere",
        "nein"
      ]
    }
  ],
  ...
}
```

```
...
{
  "name": "zink",
  "type": "float",
  "max_digits": 3,
  "decimal_places": 1,
  "max": 50.0,
  "min": 0.0,
  "comment": "Zink <kt>[µmol/l]</kt>"
},
{
  "name": "homa",
  "type": "float",
  "max_digits": 3,
  "decimal_places": 1,
  "function": {
    "name": "homa_ir_mg_dl",
    "args": {
      "insulin": "insulin_spiegel",
      "glukose": "glucose_spiegel"
    }
  },
  "max": 99.0,
  "min": 0.0,
  "comment": "HOMA-Index"
},
{
  "name": „cortisol_1mg“,
  ...
}
```

# »Single Source of Truth« (2)



# Validation with JSON Schema (1)

»**JSON Schema** is a vocabulary that allows you to **annotate** and **validate** JSON documents.«

- JSON Schema specification:

<https://json-schema.org>

- Python version:

<https://github.com/python-jsonschema/jsonschema>

```
pip install jsonschema
```

# Validation with JSON Schema (2)

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "description": "Verification of json model description files",
  "type": "object",
  "required": ["study", "model", "fields", "unique_together"],
  "properties": {
    "study": {
      "const": "btx"
    },
    "model": {
      "enum": ["calorimetry", "diseases", "lab", "psych_diseases", "psych_medication",
        "reg_medication", "seca", "skin_barrier_messung", "teilnahme", "today_medication"]
    },
    "fields": {
      "type": "array",
      "uniqueItems": true,
      "items": {
        "oneOf": [
          {
            "$ref": "file:///code/schema/schema.string.json"
          },
          {
            "$ref": "file:///code/schema/schema.integer.json"
          },
          {
            "$ref": "file:///code/schema/schema.float.json"
          },
          {
            "$ref": "file:///code/schema/schema.enum.json"
          },
          {
            "$ref": "file:///code/schema/schema.boolean.json"
          }
        ]
      }
    }
  }
}
```

```
{
  "study": "btx",
  "model": "calorimetry",
  "fields": [
    {
      "name": "pat_id",
      "type": "pat_id"
    },
    {
      "name": "visite",
      "type": "foo",
      "min": 0,
      "max": 5,
      "comment": "Visitentag, 1. Visitentag = 0"
    },
    {
      "name": "spo2_percent",
      "type": "integer",
      "min": 80,
      "max": 100,
      "comment": "Sauerstoffsättigung <kt>[%]</kt>"
    }
  ]
}
```

```
root@studydb-docker:/code/scripts# python3 validate.py
-----
directory: s_zanadio
now at table: zanadio...ok
-----
directory: s_btx
now at table: calorimetry...
Path: $.fields[1].type
Message: {'name': 'visite', 'type': 'foo', 'min': 0, 'max': 5,
'comment': 'Visitentag, 1. Visitentag = 0'}
is not valid under any of the given schemas
now at table: diseases...ok
now at table: lab...ok
now at table: psych_diseases...ok
now at table: psych_medication...ok
...
```



# Model-from-JSON (1)

- Django has an abstraction layer using »models«. One model maps to one database table (here: PostgreSQL).
- We have written a template-based Python tool to parse our own model descriptions (JSON) and generate the required Django (Python) code.



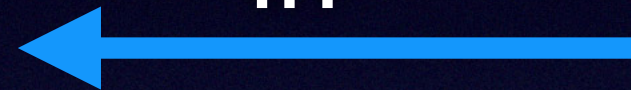
# Generating test data from JSON table definition

```
for line in range(n_lines):
    row = list()
    for jfield in cf["fields"]:
        try:
            if jfield["type"] == "pat_id":
                patid = random.randint(PATID_ID_MIN, PATID_ID_MAX)
                row.append(str(patid))
            elif jfield["type"] == "integer":
                imin = jfield.get("min", 0)
                imax = jfield.get("max", 100)
                d = random.randint(imin, imax)
                row.append(str(d))
            elif jfield["type"] == "float":
                rmin = jfield.get("min", 0)
                rmax = jfield.get("max", 100.0)
                max_digits = jfield.get("max_digits", 5)
                decimal_places = jfield.get("decimal_places", 2)
                r = random.uniform(rmin, rmax)
                row.append('{:~max}~{prec}f'
                            .format(r, max=max_digits,
                                    prec=decimal_places))
            elif jfield["type"] == "boolean":
                if random.random() > 0.5:
                    b = "ja"
                else:
                    b = "nein"
                row.append(b)
            elif jfield["type"] == "date":
                fake_date = fake.date_between_dates(
                    UPLOAD_DATETIME_START,
                    UPLOAD_DATETIME_END).strftime("%d.%m.%Y")
                row.append(fake_date)
            ...
            elif jfield["type"] == "enum":
                selected = random.choice(jfield["values"])
                row.append(selected)
```

## JSON definition for fields

```
{
  "name": "spo2_percent",
  "type": "integer",
  "min": 80,
  "max": 100,
  "comment": "Sauerstoffsättigung <kt>[%]</kt>"
},
{
  "name": "hr",
  "type": "float",
  "max_digits": 5,
  "decimal_places": 2,
  "min": 30,
  "max": 150,
  "comment": "Herzfrequenz <kt>[1/min]</kt>"
},
}
```

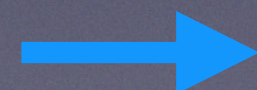
in



## CSV import for one table

```
"pat_id";"visite";"spo2_percent";"hr";"feco2_percent";"vco2";"vo2";"ve";"rer";"vo2_kg";"e
"1080";"0";"92";"131.73";"0.21";"650.38";"468.15";"146.02";"1.4";"1.1";"9563";"2693";"76"
"1063";"0";"99";"107.06";"1.52";"827.57";"262.75";"10.12";"0.9";"2.7";"9347";"1050";"58";
"1024";"0";"96";"45.36";"1.86";"640.37";"385.11";"17.25";"1.6";"8.4";"8463";"1599";"29";"
"1075";"1";"81";"133.23";"0.12";"408.41";"384.76";"143.93";"1.7";"5.5";"7208";"2489";"32"
"1095";"2";"96";"121.02";"1.82";"416.40";"318.08";"298.39";"1.0";"2.3";"2556";"4446";"57"
"1010";"3";"99";"53.14";"0.01";"264.71";"979.61";"26.54";"1.9";"6.0";"4710";"228";"80";"2
"1021";"1";"90";"61.06";"0.71";"553.12";"79.90";"126.01";"1.0";"7.1";"5066";"1873";"22";
"1017";"1";"92";"72.79";"0.88";"396.52";"913.18";"102.36";"1.4";"0.3";"3375";"1111";"54";
"1032";"3";"82";"35.15";"0.64";"901.25";"178.32";"239.52";"1.1";"7.7";"6903";"1519";"15";
```

out





Sie für jede Spielart eine Antwort: «einmal pro Woche» oder «ein bis mehr»

a	<input type="checkbox"/>	<input type="checkbox"/>	Kartenspielen
b	<input type="checkbox"/>	<input type="checkbox"/>	Pfandweine (vor Ort, über den Webshop)
c	<input type="checkbox"/>	<input type="checkbox"/>	Sportwetten
d	<input type="checkbox"/>	<input type="checkbox"/>	Würfelspiele
e	<input type="checkbox"/>	<input type="checkbox"/>	Spiele im Kassenlokal
f	<input type="checkbox"/>	<input type="checkbox"/>	Lotterien
g	<input type="checkbox"/>	<input type="checkbox"/>	Blöge um...

Logged in as: mcurie

[MENU](#)

studydb 0.14.07 of 09.05.2022

BTX table:  PatID:  Visite:

pat id	<input type="text" value="1002"/>	Def: Patienten-ID
visite	<input type="text" value="4"/>	Def: Visitentag, 1. Visitentag = 0
spo2 [%]	<input type="text" value="89"/>	<b>Begründung:</b> <input type="text" value="Messgerät defekt"/> Def: Sauerstoffsättigung [%]
hr	<input type="text" value="118.02"/>	Def: Herzfrequenz [1/min]
feco2 [%]	<input type="text" value="1.62"/>	Def: FeCO <sub>2</sub> [%]
vco2	<input type="text" value="437.84"/>	Def: Kohlenstoffdioxid-Abgabe (V'CO <sub>2</sub> ) [ml/min]
vo2	<input type="text" value="210.75"/>	Def: Sauerstoff-Aufnahme (V'O <sub>2</sub> ) [ml/min]
ve	<input type="text" value="159.43"/>	Def: Ventilation (V'E) [L/min]: Lungenbelüftung während der Atmung innerhalb einer Minute
rer	<input type="text" value="1.6"/>	Def: respiratorische Austauschrate von Kohlendioxidausstoß und Sauerstoffverbrauch (VCO <sub>2</sub> /VO <sub>2</sub> )
vo2 [kg]	<input type="text" value="0.9"/>	Def: V'O <sub>2</sub> /kg [ml/min/kg]
ee d	<input type="text" value="1407"/>	Def: Energie-Umsatz pro Tag [kcal/Tag]
cho e d	<input type="text" value="1703"/>	Def: Kohlenhydrate pro Tag [kcal/Tag]
cho [%]	<input type="text" value="13"/>	Def: Kohlenhydratgehalt in Prozent [%]

rendering with KaTeX



# GUI from JSON table definition

```

...
{
  "name": "spo2_percent",
  "type": "integer",
  "min": 80,
  "max": 100,
  "comment": "Sauerstoffsättigung <kt>[%]</kt>"
},
{
  "name": "hr",
  "type": "float",
  "max_digits": 5,
  "decimal_places": 2,
  "min": 30,
  "max": 150,
  "comment": "Herzfrequenz <kt>[1/min]</kt>"
},
...

```



- GUI generated dynamically
- »Good Clinical Practice«: existing values can only be changed when a reason is given



## Import Excel/CSV file

Filenames need to have format: `<study>_<table>[-<comment>].csv` or `.xlsx`

e.g. `btx_psych_medication.xlsx` or `btx_calorimeter-alpha.xlsx`

Excel/CSV file:  No file selected.

[Import Excel/CSV file](#)

## Results of last Import

```
"btx_calorimetry-demo.xlsx" uploaded by mcurie at 2022-05-13 01:38:43.
total lines: 3, successfully imported: 2, line(s) with errors: 1
errors to fix:
▪ [4] col: visite, value: "a"
  *** Integer numbers must only contain digits.
▪ [4] col: spo2_percent, value: "102"
  *** Value must be smaller than 100.
▪ [4] col: ve, value: "epsilon"
  *** Not a float number.
```

# JSON table definition for validation of input

```
{
  "name": "visite",
  "type": "integer",
  "min": 0,
  "max": 5,
  "comment": "Visitentag, 1. Visitentag = 0"
},
{
  "name": "spo2_percent",
  "type": "integer",
  "min": 80,
  "max": 100,
  "comment": "Sauerstoffsättigung <kt>[%]</kt>"
},
{
  "name": "ve",
  "type": "float",
  "max_digits": 5,
  "decimal_places": 2,
  "min": 0,
  "max": 300,
  "comment": "Ventilation <kt>(V'E)</kt> <kt>[L/min]</kt>: Lungenbelüftung wähn"
},
}
```

	A	B	C	D	E	F	G	H	I	J
1	pat_id	visite	spo2_percent	hr	feco2_percent	vco2	vo2	ve	rer	vo2_kg
2	1111	0	98	65	0,80	244,26	274,3	39,42	0,89	
3	2222	0	98	54	0,79	321,05	354,37	52,79	0,92	
4	3333	a	102	48	0,78	250,03	274,5	epsilon	0,91	



## Electronic Case Report Form

automatically generated at: 12.05.2022 12:58:55 by StudyDB 0.14.07, git: 360d1118

### Table of Contents

1	Study: btx, direct tables	1
1.1	Table: calorimetry (direct)	1
1.2	Table: diseases (direct)	2
1.3	Table: lab (direct)	4
1.4	Table: psych_diseases (direct)	8
1.5	Table: psych_medication (direct)	13
1.6	Table: reg_medication (direct)	17
1.7	Table: seca (direct)	21
1.8	Table: skin_barrier_messung (direct)	23
1.9	Table: teilnahme (direct)	25
1.10	Table: today_medication (direct)	28
2	Survey: 697139	35
2.1	Table: btx_basic_data (Survey: 697139)	35
2.2	Table: maia (Survey: 697139)	40
2.3	Table: iwqol_lite (Survey: 697139)	43
2.4	Table: bis_11 (Survey: 697139)	46
2.5	Table: cd_risc_25 (Survey: 697139)	49
2.6	Table: fcq_t_r (Survey: 697139)	52
2.7	Table: hads_d (Survey: 697139)	54
2.8	Table: opd_sf (Survey: 697139)	55
2.9	Table: dars_17 (Survey: 697139)	66
2.10	Table: pss_10 (Survey: 697139)	68
2.11	Table: sss_v (Survey: 697139)	69
2.12	Table: dfs (Survey: 697139)	72
2.13	Table: sci (Survey: 697139)	75
2.14	Table: spsrq (Survey: 697139)	81
2.15	Table: fev (Survey: 697139)	84
2.16	Table: ecr_rd8 (Survey: 697139)	89
2.17	Table: skin_barrier (Survey: 697139)	90
3	Survey: 818915	92
3.1	Table: srrs (Survey: 818915)	92

### 1 Study: btx, direct tables

#### 1.1 Table: calorimetry (direct)

- field: ► `pat_id`  
type: `pat_id`, MPI-SF internal proband ID; index field
- field: ► `visite`  
type: `integer`, allowed range: `[0, 5]`  
Visitentag, 1. Visitentag = 0
- field: ► `spo2_percent`  
type: `integer`, allowed range: `[80, 100]`

# eCRF from JSON table definitions (on demand)

L<sup>A</sup>T<sub>E</sub>X

### Index

- absolute\_fat\_mass (field), 22
- acanthosis\_nigricans (field), 25
- ace\_hemmer (field), 18, 29
- agomelatin (field), 15, 33
- ait (field), 3

definition of calc function: getting rid of Excel

```
def homa_ir_mg_dl(insulin_spiegel=None, glucose_spiegel=None):
    return float(insulin_spiegel) * float(glucose_spiegel) / 405.0
```

parameter mapping

enum type

- field: ► `zink`  
type: `float`, allowed range: `[0.0, 50.0]`  
Zink [mol/l]
- field: ► `homa`  
type: `float`, allowed range: `[0.0, 99.0]`  
function: `homa_ir_mg_dl`, args: `[insulin ↦ insulin_spiegel | glukose ↦ glucose_spiegel]`  
HOMA-Index
- field: ► `cortisol_1mg`  
type: `float`, allowed range: `[0.0, 99.0]`  
Cortisolwert nach 1mg Dexamethason am Vorabend (Dexa-Hemmtest) [g/l]

#### 1.4 Table: psych\_diseases (direct)

- field: ► `pat_id`  
type: `pat_id`, MPI-SF internal proband ID; index field
- field: ► `visite`  
type: `integer`, allowed range: `[0, 5]`
- field: ► `f00`  
type: `enum`, allowed values: `[f00.0 | f00.1 | f00.2 | f00.9 | nein | andere]`  
Hat der Proband eine F00 Demenz bei Alzheimer? Nach ICD-10-GM

# Excel Add-In for Manipulating Points on Charts (MPOC)

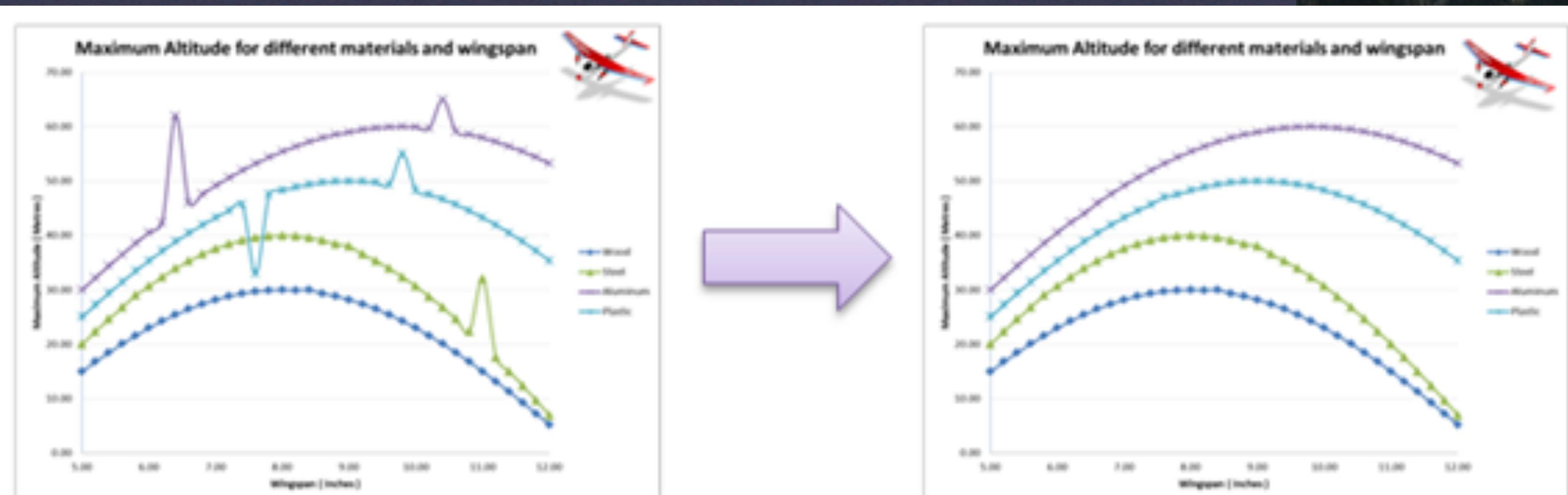
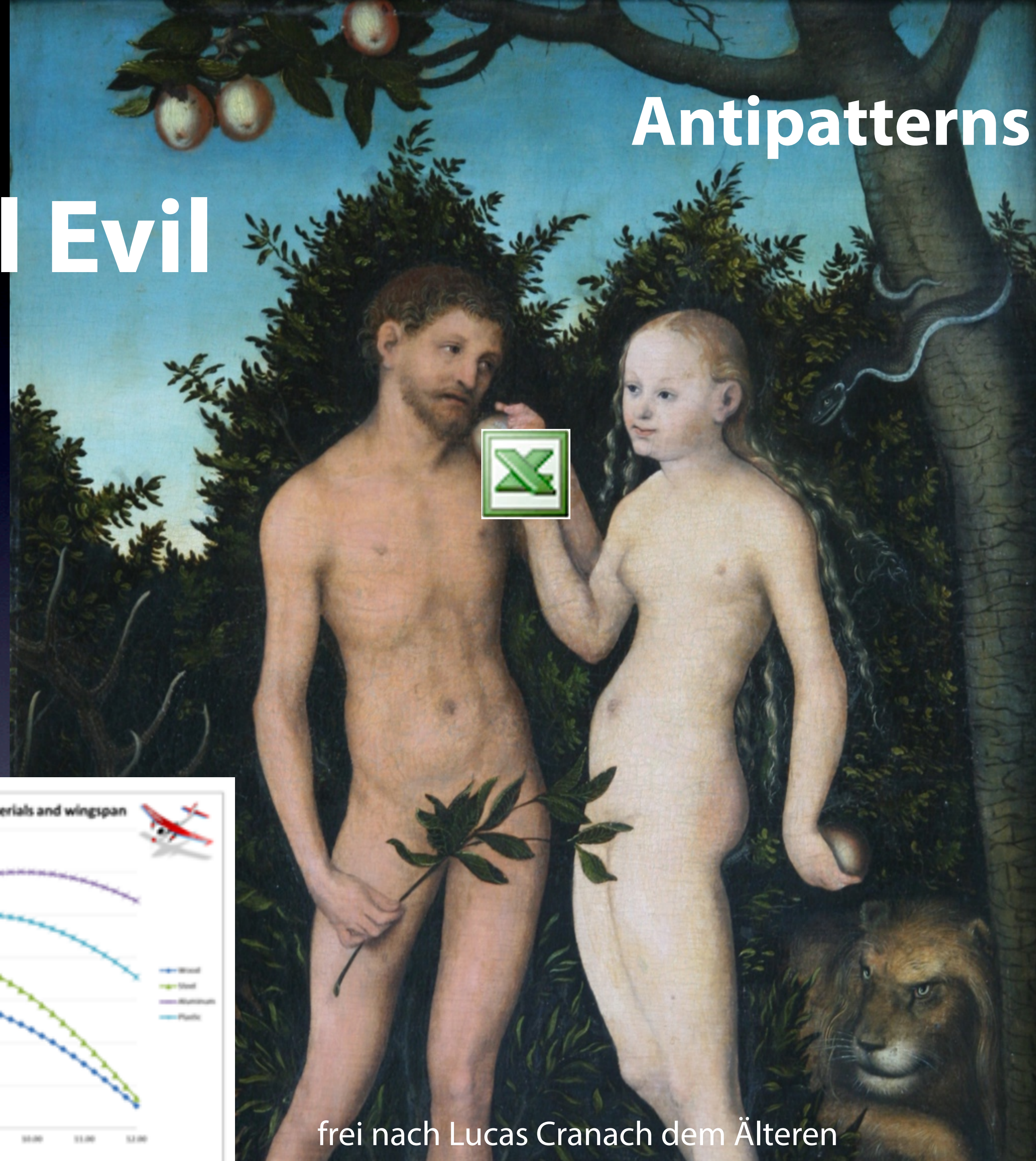
by [Diego Oppenheimer](#)  
on November 02

## Overview

In Excel 2007, the ability to directly resize or reposition points on the chart was deprecated. This feature was sometimes referred to as "**Graphical Goal Seek.**" [...] The scientists are elated at their find but these are some erroneous values in the data (show with red circles on the chart). The field scientist tells us that the erroneous readings were caused by some loose wiring to the altimeter used in the experiment. Since the trends are pretty obvious, **the research department wants to clean up the charts before presenting their findings to the management.**

# Excel Evil

# Antipatterns



# LimeSurvey (Questionnaires) (1)

Max-Planck-Institut für Stoffwechselforschung

MoCo-D 2nd

0%

In dieser Umfrage sind 29 Fragen enthalten.

Weiter

Zwischengespeicherte Umfrage laden

© 2021 MPI für Stoffwechselforschung

Erhebung der Händigkeit nach Oldfield (1971)

Bitte kreuzen Sie an, welche Hand Sie bei folgenden Aktivitäten benutzen.

\*Welche Hand benutzen Sie beim Schreiben?

Bitte wählen Sie eine der folgenden Antworten:

linke Hand

rechte Hand

beide Hände gleich häufig

\*Benutzen Sie beim Schreiben auch manchmal die andere Hand?

Bitte wählen Sie eine der folgenden Antworten:

Ja, manchmal benutze ich die andere Hand als die eben angegebene.

Nein, ich benutze immer die eben angegebene Hand.

\*Welche Hand benutzen Sie beim Malen?

Bitte wählen Sie eine der folgenden Antworten:

linke Hand

rechte Hand

beide Hände gleich häufig

\*Benutzen Sie beim Malen auch manchmal die andere Hand?

Bitte wählen Sie eine der folgenden Antworten:

Ja, manchmal benutze ich die andere Hand als die eben angegebene.

Nein, ich benutze immer die eben angegebene Hand.

Questionnaire as seen by patient (left), PDF representation of answers (right).

**Teil B: Rechts- / Links-Haendigkeit**

Erhebung der Händigkeit nach Oldfield (1971) Bitte kreuzen Sie an, welche Hand Sie bei folgenden Aktivitäten benutzen, wenn Sie beide Hände etwa gleich häufig benutzen, wählen Sie bitte "beide".

**B1. Welche Hand benutzen Sie beim Schreiben?**

linke Hand

rechte Hand

beide Hände gleich häufig

**B2. Benutzen Sie beim Schreiben auch manchmal die andere Hand?**

Ja, manchmal benutze ich die andere Hand als die eben angegebene.

Nein, ich benutze immer die eben angegebene Hand.

**B3. Welche Hand benutzen Sie beim Malen?**

linke Hand

rechte Hand

beide Hände gleich häufig

**B4. Benutzen Sie beim Malen auch manchmal die andere Hand?**

Ja, manchmal benutze ich die andere Hand als die eben angegebene.

Nein, ich benutze immer die eben angegebene Hand.

- Questionnaire on LimeSurvey server is being prepped remotely by StudyDB.
- Questionnaire is being filled out by patient.
- LimeSurvey generates »QueXML-PDF« (extra level of quality assurance for documentation) and CSV files in download area.
- LimeSurvey signals StudyDB that a finished questionnaire (»survey«) is ready to download, messaging uses ZeroMQ (MPI-SF adaption).



# LimeSurvey: Timestamping finished questionnaire data

local ZMQ server log: received message from LimeSurvey server

```
$ var/log$ tail zmq.log
2022-02-06 14:07:56 t: d6327c2ab8920f6e06 survey_completed: sid:325136
2022-02-06 14:07:56 survey t: d6327c2ab8920f6e06 sid: 325136
2022-02-06 14:07:57 [zmq-trigger-surveyprocessing] creating queue entry.
```

fetch questionnaire data (csv, pdf) from LimeSurvey server (rsync)

```
$ var/log$ tail survey-trans.log
===== 2022-02-06 14:07:58
receiving incremental file list
./
lxmpisf-325136-d6327c2ab8920f6e06.csv
lxmpisf-325136-d6327c2ab8920f6e06.pdf
sent 119 bytes received 4,486,610 bytes 2,991,152.67 bytes/sec
total size is 4,485,324 speedup is 1.00
```

import files, calculate SHA hashes

```
$ var/log$ tail survey.log
2022-02-06 14:07:58 [process_new_surveyfiles]
2022-02-06 14:07:58 [transfer_files_from_limesurvey]
2022-02-06 14:07:59 [transfer_files_from_limesurvey] done.
2022-02-06 14:07:59 [process_transferred_survey_files]
2022-02-06 14:08:00 <imported>
survey_root_name: "lxmpisf-325136-d6327c2ab8920f6e06"
csv_hash: "7e3b3a5cfe5d09972043afdb8a21b1429c69886161...775b4baab7"
pdf_hash: "95c7b2b8bffffb97bedbac844a867ea7d786b7c96...3505b804d9"
tlog_hash: "728223d279def915da8b41eec15facd5ff9b833dc8...4b842981f9"
2022-02-06 14:08:00 [process_transferred_survey_files] done in: 1.032s
2022-02-06 14:08:00 [process_new_surveyfiles] done.
```

create ».tlog« summary files (containing hashes)

```
$ uploads/survey/2022-KW05$ more lxmpisf-325136-d6327c2ab8920f6e06.tlog
studydb 0.07.01 at 2022-02-06 14:08:00.
patid: "114"
survey: "325136", token: "d6327c2ab8920f6e06"
csv_filename: "lxmpisf-325136-d6327c2ab8920f6e06.csv"
csv_shash: "7e3b3a5cfe5d09972043afdb8a21b1429c6988616169408...aab7"
pdf_filename: "lxmpisf-325136-d6327c2ab8920f6e06.csv"
pdf_shash: "95c7b2b8bffffb97bedbac844a867ea7d786b7c96e86d3...04d9"
```

How to Timestamp (RFC3161) - keep ».tsr« files for validation

```
openssl ts -query -data <file> -cert -sha256 -no_nonce -out <file>.tsq

curl -s -S -k -H "Content-Type: application/timestamp-query"
--data-binary @<file>.tsq "http://zeitstempel.dfn.de" -o <file>.tsr
```

survey »325136« with token »d6327...« after timestamping:

```
$ uploads/survey/2022-KW05$ ls -lt
...
studydb-user 6616 Feb 6 14:08 lxmpisf-325136-d6327c2ab8920f6e06.tlog.tsr
studydb-user 59 Feb 6 14:08 lxmpisf-325136-d6327c2ab8920f6e06.tlog.tsq
studydb-user 373 Feb 6 14:08 lxmpisf-325136-d6327c2ab8920f6e06.tlog
studydb-user 516 Feb 6 14:07 lxmpisf-325136-d6327c2ab8920f6e06.csv
studydb-user 4484808 Feb 6 14:07 lxmpisf-325136-d6327c2ab8920f6e06.pdf
...
```

validation in two steps (validation is offline and standalone)

```
# Validation 1 (does <file> match the timestamped hash?):
openssl ts -verify -data <file> \
-in <file>.tsr \
-CAfile dfn-chain.txt -untrusted dfn-crl.txt
...
Verification: OK

# Validation 2 (what is the datetime signed by the timestamping service?):
openssl ts -reply -in <file>.tsr -text
...
Serial number: 0xE73C740959B8070BC98
Time stamp: Feb 6 13:08:00 2022 GMT
```

simple: just  
two CLI calls

# Timestamping Overview

- Motivation: given a data set (raw data file), we want to **prove that this file was in a specific state (hash value) at a specific time**. If you have many files, you could hash them and timestamp a report file that lists those hashes.
- We need an external agent for this and have considered a blockchain (Bloxberg) or using a **timestamping service (RFC 3161)**.
- For our use case, RFC 3161 has a number of advantages:
  - (1) we can immediately use one or more timestamping servers (e.g. »Zeitstempel« by DFN), many are freely available,
  - (2) usage on **Linux works out-of-the-box** with just two CLI calls (very fast),
  - (3) **validation is completely independent** of any network service and standalone (PKI based).
- RFC 3161 requires that all parties trust the external timestamping agency (primarily DFN, in our case; in particular, validation requires that the external timestamping agency's PKI infrastructure is not compromised).
- [https://en.wikipedia.org/wiki/Trusted\\_timestamping](https://en.wikipedia.org/wiki/Trusted_timestamping)
- <https://www.pki.dfn.de/faq-zeitstempel>

# Optimized table viewer

StudyDB

Max Planck Institute for Metabolism Research

Logged in as: mcurie

studydb 0.14.07 of 09.05.2022

BTX table: calorimetry

data meta marker Excel CSVY meta (PDF)

found: 180 [1, 25] at 13.05.2022 10:50:03 row: 1001;3 col: hr

pat id	visite	spo2 [%]	hr	feco2 [%]	vco2	vo2	ve	rer	vo2 [kg]	ee d	cho e d	cho [%]	fat e d	fat [%]	pro e d	pro [%]	rr syst	rr diast
1000	1	89	43.86	1.47														
1000	3	81	119.71	1.19														
1000	5	87	77.18	0.75														
1001	3	84	51.06	1.25														
1001	4	90	44.35	0.95														
1001	5	100	45.58	0.36														
1002	1	84	116.87	0.65														
1002	3	90	104.96	0.97														
1002	4	89	118.02	1.62														
1005	4	88	114.99	0.95														
1006	1	84	137.40	0.54														
1006	5	94	47.59	1.53														
1007	4	97	131.90	1.70														
1008	2	82	134.58	1.25	132.09	500.99	260.47	1.7	6.5	8386	2126	84	470	56	4937	28	164	134

BTX table: calorimetry

data meta meta (PDF)

name	type	comment	values	min	max	max_digits	decimal_places
pat_id	PAT_ID						
visite	INTEGER	Visitentag, 1. Visitentag = 0		0	5		
spo2_percent	INTEGER	Sauerstoffsättigung [%]		80	100		
hr	FLOAT	Herzfrequenz [1/min]		30	150	5	2
feco2_percent	FLOAT	FeCO <sub>2</sub> [%]		0	2	3	2
vco2	FLOAT	Kohlenstoffdioxid-Abgabe (V'CO <sub>2</sub> ) [ml/min]		0	1000	6	2
vo2	FLOAT	Sauerstoff-Aufnahme (V'O <sub>2</sub> ) [ml/min]		0	1000	6	2
ve	FLOAT	Ventilation (V'E) [L/min]: Lungenbelüftung während der Atmung innerhalb einer Minute		0	300	5	2
rer	FLOAT	respiratorische Austauschrate von Kohlendioxidausstoß und Sauerstoffverbrauch (VCO <sub>2</sub> /VO <sub>2</sub> )		0.60	2	2	1

rendering with KaTeX

## performance tuning (Django)

```
# m: model instance
m = apps.get_model(app_name, model_name)
# ... get list of rows for page
# iterate over fields in row
value = row.__dict__.get(field_name)
# avoid this here:
# value = getattr(row, field_name)
```

You can navigate using the arrow keys (keyboard); you might need to scroll (right, down) when using the mouse.