THURGAUER WIRTSCHAFTSINSTITUT
an der Universität Konstanz

Universität Konstanz

Scheduling with z-Tree 5.*

…and other stuff introduced in this version

z-Tree

Main new features in z-Tree 5

- More that one treatment can run at the same time, i.e. different subjects can be in different treatments.
- Treatments can be scheduled from within a treatment. This allows automated session.
  - Treatments with zero subjects can be used for this purpose.
  - Treatments can have an unspecified number of subject, i.e. such treatments can run with any number of subjects.
- It is possible to specify a treatment that runs when a client is started – a welcome treatment.
- It is possible to kick subjects out of a treatment (dropouts) – and guide them into another treatment.

Potential problems for users of V4.*?

z-Tree

- In general: NO
- The session table is renamed to clients.
  - This is automatically done, you only have to get used to it.
- New tables: treatments, participations, sessionglobals, clients.
  - If you uses tables with this names, you have to rename them.
  - There is procedure to support this.
- New keywords:
  - treatment, welcome, dropout , attime , when, whenever , afterstart , afterend , select , sortedby , using, finishintreatments , clientrelease , clientremove , clientreplace
  - These keywords cannot be uses as variable names
- New variables in tables
  - clients (session):ClientNumber, ClientName, TreatmentNumber, StateString, RemainingSeconds
  - globals: Repetition
  - subjects: ClientNumber
- Some variables are now read only (e.g. Subject,…), and in some tables yo cannot add records (subjects, globals,…)

Treatment with unspecified number of subjects

- If the number of subjects is set to -1, the treatment can be used for any number of subjects.
- If the number of subjects is set to 0, the treatment ist started with no subjects. This option is useful for scheduling (see later)

Multiple treatments run at the same time

Clients' Table

8 clients	client name	selected	treatment	state	time	stop after period
1	1	[ ]	37	*** Stage ***	6	[ ]
2	2	[ ]	---	Ready	-	[ ]
3	3	[ ]	---	Ready	-	[ ]
4	4	[ ]	36	*** Stage ***	3	[ ]
5	5	[ ]	36	*** Stage ***	3	[ ]
6	6	[ ]	36	*** Stage ***	3	[ ]
7	7	[ ]	37	*** Stage ***	6	[ ]
8	8	[ ]	37	*** Stage ***	6	[ ]

- In the Connection Monitor, clients can be selected and then the treatment can be started.
- This is also possible when there is already a treatment running.
- Without selection, all clients are assigned to the treatment.
- The fourth columns shows the treatment number the client is currently in.

Clients' Table

8 clients	client name	selected	treatment	state	time	stop after period
1	2	[x]	---	Ready	-	[ ]
2	3	[ ]	2	*** Stage ***	14	[ ]
3	4	[ ]	2	*** Stage ***	14	[ ]
4	1	[ ]	3	*** Stage 2 ***	31	[ ]
5	5	[ ]	3	*** Stage 2 ***	31	[ ]
6	6	[x]	---			
7	7	[ ]	---			
8	8	[ ]	---			

Multiple treatments run at the same time - Applications

- Different treatments in one session.
- Different subjects/group continue at own pace.
- Separate markets for efficiency.
- …dropout (details below)

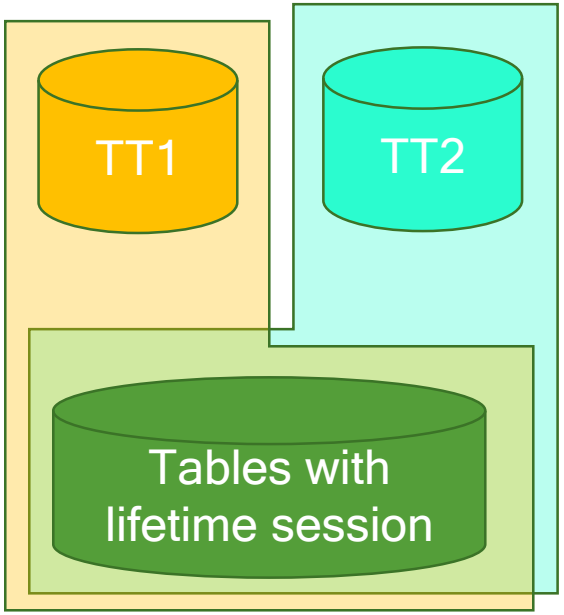
Multiple treatments run at the same time – data structure

- In the clients table:
  - ClientNumber: ID of the client. It is assigned when the client gets in the first treatment.
  - ClientName: Name of the connnection (/name *x* in z-Leaf)
  - TreatmentNumber: The number of the treatment, this client is in (or -1).
  - Subject: The subject ID that this client has in the treatment (or -1).
  - StateString: String of the state.
  - RemainingSeconds: Time as in Connection Monitor.
- In the globals table
  - TreatmentID: ID of the treatment that is running (1 is first defined TT).
  - TreatmentNumber: Number of the treatment that is running (1 is first run TT).
- In the subjects table
  - ClientNumber: ID in the clients table.
  - Subject: ID in the treatment.

Multiple treatments run at the same time - details

z-Tree

- Different treatments share the tables with lifetime session.
- It is possible to share data between the treatments using tables with lifetime session.
- However, the treatments do not actively exchange information.
  - Information in a session table that is changed by one treatment is not sent to the subjects in another treatment. Be careful and do not rely on anything. Rules in this respect are subject to change.
- It is not possible to run a treatment with a table of lifetime session when another treatment runs the same table not in lifetime session, and vice versa.



Scheduling

- You can start treatments that run…
  - …when a new client connects: welcome treatment.
  - …at or after some time.
  - …when another treatment starts or ends.
  - …when a client drops out of a treatment: dropout treatment.
- You can kick subjects out of a treatment…
  - …and send it to another treatment

The treatments table

- The treatments table contains the treatments (and questionnaires) that are currently running, that did run and are planned to be run.
- Variables
  - TreatmentID: ID of the treatment; set when record is created.
  - BaseTreatment: ID of treatment of which this is a copy.
  - StartMethod: how it has been started.
  - FileName: file name of name of window if not yet saved.
  - BackupFileName: file name of backup file (with path).
  - NumberOfSubjects: number of subject that have been started in the TT.
  - TreatmentNumber: Counter of the started treatments; -1 if not yet started.
  - StartTimeString: start time or empty if the treatment has not yet been started.
  - EndTimeString: end time or empty.
  - ErrorString: error message (deactivated the entry)
  - MessageString: message.
  - EmptyString: just empty; allows increasing the width of the MessageString column.

How to schedule treatments

- You can start treatment from the menu.
  - It is checked...
    - …that the treatment has no errors.
    - …that the number of selected subjects matches the number specified in the treatment.
    - …that all clients that should be started are in state Ready.
  - The entry in the treatments table is made.
  - The treatment starts.
- You can start treatment from a program using the `treatment` command…

The treatment command

- The treatment command allows specifying a treatment that is scheduled. The specification has the structure

  T2 = treatment(*filename*) {

  under_what_conditions

  who_participates

  post_processing

  }

- The treatment stored in filename is loaded.
- The entry is made in treatments table. The command returns the ID if the treatment.
- The treatment is started if and/or when the conditions are satisfied.
- Treatment commands can be positioned at an place. Usually, you use a treatment with zero subjects and put the specification into a program in the globals table.
- Filename and the value (T2) are retrieved/stored in the table in which the specification is placed.

Welcome treatment

z-Tree

```
T1 = treatment( "welcome.ztt" ) {
  welcome();
}
```

- If such a treatment is specified, then for every newly connected client, this treatment is independently started.
- Thus, this treatment has to have exactly one subject (or -1).
- There is only one welcome treatment. A new welcome treatment specification replaces an old one.
- Since the entry in the clients table is only created when a treatment is started, it is strongly recommended to use a welcome treatment in order to make the client visible to the programs.
- Scheduling a treatment with empty file name removes the welcome treatment.

Start at a time

z-Tree

```
T1 = treatment( "tt.ztt" ) {
  attime("20-07-19T20:09:12.22");
}
```

- The treatment will start on July 19th, 2020 at 20:09:12.22.
- The following formats are allowed:
  - In the date, one can omit year or month.
  - Time can omit minutes and seconds.
  - If T is not specified then it is assumed that it is a time without date.
- "+d:h:m:s" is d days, h hours, m minutes and s seconds after the program ist executed.
  - Day, hour and minute can be omitted.
  - “+70:90" is ok and means 70* 60 +90 seconds.

Start conditions

```
T1 = treatment( "tt.ztt" ) { // or any other filename
  attime( timestring );
  when ( condition );
  if( condition );
  whenever( condition ); // a copy of the treatment is started
}
```

- “when” condition is tested. The condition is executed in the treatment record. If it fails, the treatment waits.
- If “if” conditions fail, then the treatment is deactivated.
- Test are conducted sequentially. As soon as an “if” condition is reached it has to succeed.
- “whenever” is like “when” but a copy of the treatment is started and whenever the condition is met again, another treatment starts.

Select the clients


```
T1 = treatment( "tt.ztt" ) { // or any other filename
  attime(“+0"); // now
  preselectionprogram {
    clients.do{
      TT3 = if( ClientNumber>=3 &ClientNumber<=5 ,1,0);
    }
  }
  select (TT3 ==1) sortedby ( -ClientNumber);
}
```

- preselectionprogram is executed in the treatments record.
- Client are selected according to condition and sorted according to expression (low values first).
- If no selection is specified, all clients in he clients table start (not the clients in the Connections Monitor).

Transmitting information to the treatment using donowintreatment

- You can use donowintreatment to transmit information from the program that specifies a treatment to the specified treatment.
- This is necessary because the calling program (or even treatment) is no longer available when the specified treatment starts or is tried to start.

- `donowintreatment { program }`
- The program is executed when the treatment is specified.
- The donowintreatment program runs in the new record of the treatment table, within the scope environment of the where treatment is called.
- So if the treatment specification is in the subjects table, the scope is
  - treatment -> subjects -> globals

Consecutive start, start conditional on other treatments

```
T1= treatment ( "tt3.ztt" ) { attime("+0"); // now
 preselectionprogram {
   TT3 = if( ClientNumber>=3 &ClientNumber<=5 ,1,0);
 }
 select (TT3 ==1) sortedby ( -ClientNumber);
}

T2= treatment ( "tt3.ztt" ) {
  afterstart(T1);
  preselectionprogram {
    TT2 = if( ClientNumber==1| ClientNumber==7 | ClientNumber==8 ,1,0);}
  }
  select (TT4 ==1);
}

T3= treatment ( "tt3.ztt" ) {
  afterend(T1);
  preselectionprogram {
    TT3 = if( ClientNumber>=4 &ClientNumber<=6 ,1,0);
  }
  select (TT4 ==1) sortedby ( -ClientNumber);
}
```

- A treatment can be started conditional in whether another treatment has been started or ended.
- Note that the triggered treatments (by welcome or dropout) have another ID than the triggering treatments. Therefore, “afterend” cannot be used to do follow up “welcome” treatments. Use “when” or “whenever” in this case.

Dropout

- Execute the following program in the subjects table.

```
clientremove();
```

- Then this subjects is relleses in the session table and in the clients' table, and the ClientNumber is set to -1.
- If there is a dropout handling treatment, this client moves into this treatment, see below.
- This means that further profits are not transmitted.

Dropout

z-Tree

ClientNumber	ClientName	TreatmentNumber	Subject
1	ant1	-1	-1
2	ant2	2	2
3	ant3	2	1
4	ant4	-1	-1
5	an5	2	3

Subject	ClientNumber	TreatmentNumber
1	3	2
2	2	2
3	5	2

clientremove

Subject	ClientNumber	TreatmentNumber
1	3	2
2	-1	-1
3	5	2

Dropout handling

- Use the following program.

```
T1= treatment ( "tt5S-dropout.ztt" ) {
  attime("+0");
}

T2 =  treatment ( "dropout.ztt" ) {
  dropout( T1 );
}
```

- The treatment `dropout.ztt` runs for any client that drops out.
- Scheduling a treatment with empty file name removes the dropout treatment … for the respective treatment.

Dropout payoff

- Default: Profit is transferred to session
- Delete payoff of the current treatment:
  - Profit = -if(Period ==1,0, OLDsubjects.find(same(Subject),TotalProfit)) ;
  - clientremove();
- Delete payoff of all treatments
  - Profit = -if(Period ==1,0, OLDsubjects.find(same(Subject),TotalProfit))
  - - session.find(same(ClientNumber), MoneyEarned);
  - clientremove();

Summary of all options

- welcome
- dropout
- attime
- if
- when
- whenever
- afterstart
- afterend
- donowintreatmentprogram
- preselectionprogram
- select
- sortby
- posttreatmentprogram

Example of public good, stranger

- Dropout recognition: no input

```
// dropout handling

if ( Contribution ==-1 & Out == 0 ){
    Out = 1;
    clientremove();
}
```

Example of public good, stranger

- Group assignment in profit stage

```
S = subjects.sum(Subject);

repeat {
  subjects.do{  r=2*Out +random(); }
  subjects.do{  Rank = count( r >=:r ); }
} while ( S != subjects.sum(Rank ));

N = subjects.count();
subjects.do{
   Group = roundup( Rank/ GroupSize, 1);
}
subjects.do{
   Suspended = if( count( same( Group) & Out ==1)>0,1,0);
}
AllSuspended = if( subjects.count( Suspended ==0) ==0,1,0);
```

Interaction between treatments

- In order to transmit information to further treatments, the treatment command can be complemented with program that are executed at the end of the treatment.

```
T1= treatment ( "welcome.ztt" ) {
     welcome();
    posttreatmentprogram{ WhateverYouLike = 2; }
}
```

- These program are executed in the treatments table in the record of the treatment that ends.

Example: Conditional follow up

- What you want to do after a dropout might be depend on the reason of the dropout.
- So you might want to reschedule a subject into different further treatment.
- In order to transmit information to the next treatment, you can use donowintreatment:
- In the dropout treatment

```
if (subjectsfault ) {
  T1 = treatment( "dropout_mo_payment.ztq“ )
     afterend( treatments.find( same( TreatmentNumber ), TreatmentID));
     donowintreatment{
       MyClient = : ClientNumber;
     }
     select ( ClientNumber == : MyClient );
  }
  else….
```

When and where are expressions and program executed?

- X (now:local) = treatment(now:local) {
- attime(now:local);
- afterstart(now:local);
- afterend(now:local);
- when(scheduling: treatments table);
- if(scheduling:treatments table);
- donowintreatment(now: local + new treatments record)
- preselectionprogram{ scheduling: treatments table}
- select(scheduling:session table) sortby(scheduling: session table);
- posttreatmentprogram{ finish: treatments table }
- }
- Some specifications are evaluated when and where the specification is made (now:local).
- Other specifications are evaluated later…
  - …when an attempt to schedule is made (later)
  - …when the sheduled treatment finished (finish)

Further changes

- New tables
- New variable

New tables

- treatments: scheduling information
- clients: information about the clients (replaces session table; session is automatically renamed to clients)
- participations: which client participated in which periods of which treatment (documents dropout): ParticipationID, ClientNumber, Subject, TreatmentNumber, StartInPeriod, FinishedPerid (in questionnaire 0/1)
- sessionglobals: a table with lifetime session and exactly one record
- Questionnaire has access to the following tables: clients (default), sessionglobals, treatments, participations

New info in Tables

- State and remaining time can be accessed in the clients table.
- ClientNumber in subjects table
- TreatmentID in globals table
- globals variable Repetition (1,2,…) set if you use RepeatTreatment
- added checks that ensure that system variables are not modified.

Data storage

- Backup file name (@123.ztt) more informative
- Questionnaires…
  - output is also stored in xls file, and “Separate Table…” automatically creates a table with all the questions.
  - Address form is displayed whenever a field (first name … email) is nonempty.
  - Address info is appended to the address file if there are multiple address screens.
  - Subjects are individually ready when they have finished the questionnaire.
  - Added special Payment File; sorted by name and without clientID

Smaller improvements

- Leave Stage from menu leaves the stages that are relevant when you invoke the command. When the program continues, then the stages that are now reached will not be affected by this command.
- Option /tablefontsize for server. It is useful for high resolution screens.
- Command line option treatmentdir.
- "/tree" removed in treatment menu
- “Open…”dialog allows selecting multiple treatments and questionnaires.
- Utility to “Rename Table”. Click on the table and choose the command

Better Restore procedure

- “Reload database” renamed to “Restore session”
- restore brings clients at situation when z-Tree was left, not only at the beginnig of the period.
- Gamesafe stores all database changes. It is easier to use now.
- New Command “Save All Tables”
- No more menu “Replay Leaf From GameSafe” (no longer necessary)

Changes in Tools

- Separate tables makes also a questionnaires file
- "Join questionnaires file…" replaces "Join *.sbj file…"
- Removed "Transpose *.sbj file…"

Appendix

Output files

- Files
- DATE_sessioninfo.txt: dirs & session name; written at the beginning
- DATE_clients.txt: client list + address info; regularly replaced
- @lastclt.txt: File that is stored with store client order.
- DATE_scheduler.gsfs: scheduler info, binary;
- DATE_treatments.gsft: binary of treatments table; regularly replaced
- *123.ztt: treatments; added when started
- DATE.xls //
- DATE.gsf // DBReplace and DBModify messages

- DATA.pay:
- DATA_combo.pay:
- DATA.adr:

To come

- To come later (…perhaps)
  - clientreplace()
  - Stop time for only one treatment.
  - Functions
  - Mouse tracking