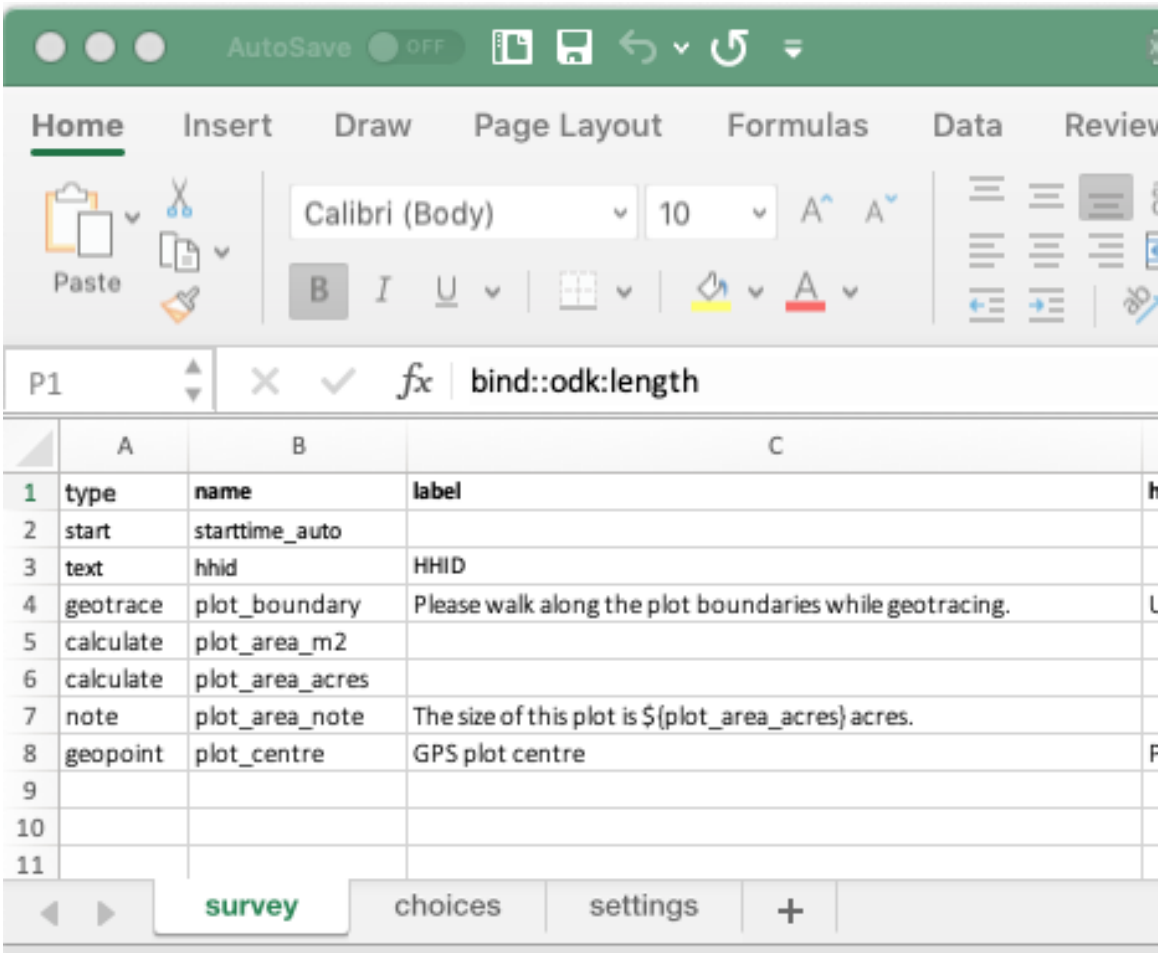
ODK: Avoiding truncation of string data (e.g. GPS data of type geotrace)

In ODK Aggregate you can supply an attribute to the to specify the maximum number of characters used to store a string field; *by default, the datastore layer limits strings to 255 characters or less*. If a submission has data beyond this length, it is silently truncated upon submission.

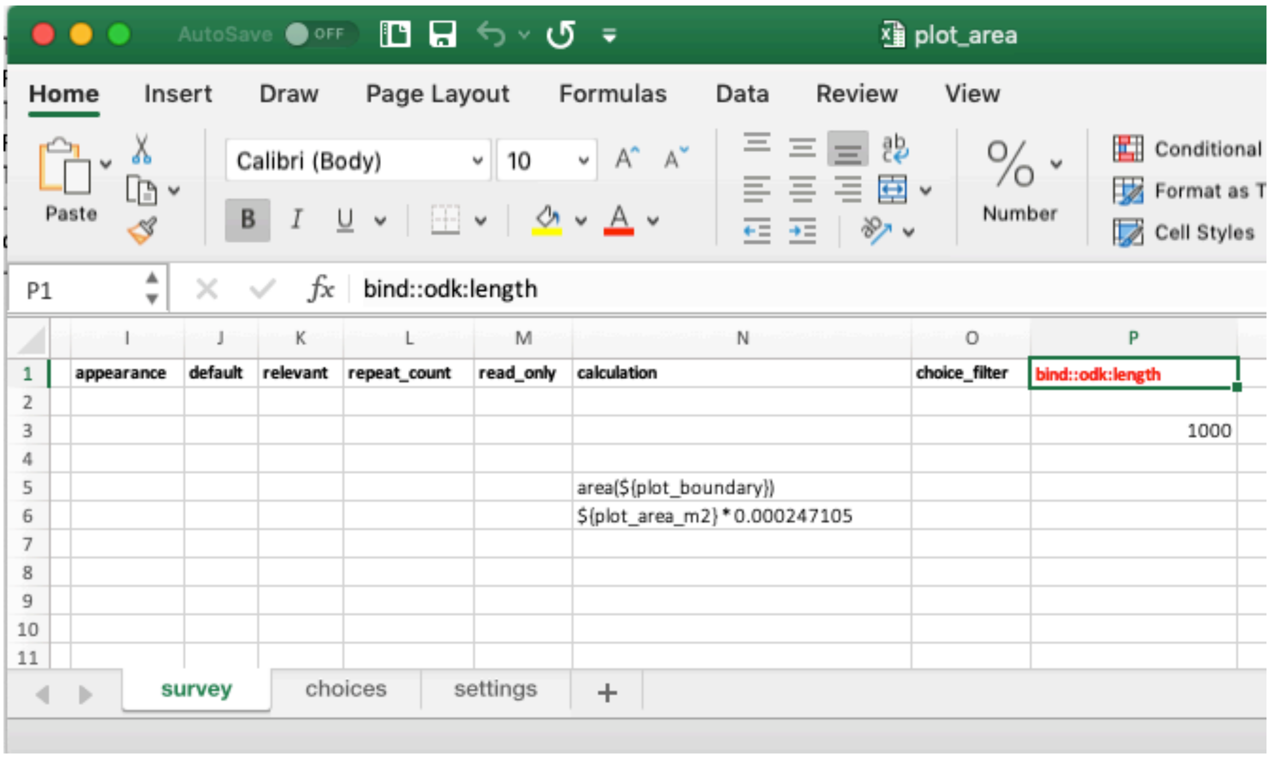
In Excel define an bind::odk:length attribute to specify the length of a string field. Length can be specified for barcode, select1 and string fields. Multi-selects (select) fields have a 255 character maximum length for each selection value (and selection values must not contain whitespace); there are no limits to the number of choices that may be selected within a multi-select.

In the example below, the attribute was added as an extra column and a value of 1000 was assigned to a variable call HHID of *ODK* type *geotrace* in row 3 (see screen shots below):



P1 fx bind::odk:length

	A	B	C
1	type	name	label
2	start	starttime_auto	
3	text	hhid	HHID
4	geotrace	plot_boundary	Please walk along the plot boundaries while geotracing.
5	calculate	plot_area_m2	
6	calculate	plot_area_acres	
7	note	plot_area_note	The size of this plot is ${plot_area_acres} acres.
8	geopoint	plot_centre	GPS plot centre
9			
10			
11			



P1 bind::odk:length

	I	J	K	L	M	N	O	P
1	appearance	default	relevant	repeat_count	read_only	calculation	choice_filter	bind::odk:length
2								
3								1000
4								
5						area(${plot_boundary})		
6						${plot_area_m2}*0.000247105		
7								
8								
9								
10								
11								

After creating a *xlm* file by using the online or offline version of the ODK XLSForm software and adding the form in ODK Aggregate, Checking the MySQL table (see screen shot below), shows that the maximum string size of the HHID variable is now 1000 as *varchar(1000).*

```
mysql> describe PLOT_AREA_CORE;
+--------------------------+----------------+
| Field                    | Type           |
+--------------------------+----------------+
| _URI                     | varchar(80)    |
| _CREATOR_URI_USER        | varchar(80)    |
| _CREATION_DATE           | datetime(6)    |
| _LAST_UPDATE_URI_USER    | varchar(80)    |
| _LAST_UPDATE_DATE        | datetime(6)    |
| _MODEL_VERSION           | int(9)         |
| _UI_VERSION              | int(9)         |
| _IS_COMPLETE             | char(1)        |
| _SUBMISSION_DATE         | datetime(6)    |
| _MARKED_AS_COMPLETE_DATE | datetime(6)    |
| PLOT_CENTRE_LAT          | decimal(38,10) |
| STARTTIME_AUTO           | datetime(6)    |
| PLOT_AREA_ACRES          | varchar(255)   |
| PLOT_CENTRE_ACC          | decimal(38,10) |
| PLOT_CENTRE_LNG          | decimal(38,10) |
| PLOT_BOUNDARY            | varchar(255)   |
| HHID                     | varchar(1000)  |
| META_INSTANCE_NAME       | varchar(255)   |
| PLOT_CENTRE_ALT          | decimal(38,10) |
| PLOT_AREA_M2             | varchar(255)   |
| PLOT_AREA_NOTE           | varchar(255)   |
| META_INSTANCE_ID         | varchar(255)   |
+--------------------------+----------------+
```

Some remarks regarding the use of VARCHAR in MySQL

MySQL `VARCHAR` is the variable-length string whose length can be up to 65,535.
MySQL stores a `VARCHAR` value as a 1-byte or 2-byte length prefix plus actual data.

The maximum length, however, is subject to maximum row size of 65,535 bytes and the character set used. It means that the total length of all columns should be less than 65,535 bytes!

If you create a new table that has two columns **`s1`** and **`s2`** with the length of 32765(+2 for length prefix) and 32766 (+2).Note that 32765+2+32766+2=65535, which is the maximum row size.
The statement created the table successfully.

However, if we increase the length of the `s1` column by 1.
MySQL will issue the error message:

```
Error Code: 1118. Row size too large. The maximum row size for
the used table type, not counting BLOBs, is 65535. This
includes storage overhead, check the manual. You have to
change some columns to TEXT or BLOBs 0.000 sec
```

The internal representation of a MySQL table has a maximum row size limit of 65,535 bytes, even if the storage engine is capable of supporting larger rows.

However, BLOB and TEXT columns only contribute 9 to 12 bytes toward the row size limit because their contents are stored separately from the rest of the row.
As far as I know, BLOB and TEXT columns are not supported by ODK as user defined field types.

So, if the row size of a table exceeds the maximum row size limit, you might consider (perhaps also in a future version of ODK) to use BLOB or TEXT columns to avoid this.

Some remarks regarding the use of Blob and Text columns

A BLOB is a binary large object that can hold a variable amount of data. The four BLOB types are TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB. These differ only in the maximum length of the values they can hold. The four TEXT types are TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT.
BLOB values are treated as binary strings (byte strings).
TEXT values are treated as nonbinary strings (character strings).

BLOB and TEXT differ from VARBINARY and VARCHAR in the following ways:

- For indexes on BLOB and TEXT columns, you must specify an index prefix length. For CHAR and VARCHAR, a prefix length is optional.
- BLOB and TEXT columns cannot have DEFAULT values.

MySQL Connector/ODBC defines BLOB values as LONGVARBINARY and TEXT values as LONGVARCHAR.

The maximum size of a BLOB or TEXT object is determined by its type, *but the largest value you actually can transmit between the client and server is determined by the amount of available memory and the size of the communications buffers.* You can change the message buffer size by changing the value of the max_allowed_packet variable, but you must do so for both the server and your client program. For example, both **mysql** and **mysqldump** enable you to change the client-side max_allowed_packet value.

Checking the row size

Suppose a table called PLOT_AREA_CORE is part of the database *odkgeneral*, then use the commands below to calculate its row size:

mysql> use odkgeneral;

mysql> select sum(character_maximum_length) from information_schema.columns where table_name = 'PLOT_AREA_CORE';

Output:

```
+-----------------------------------------------+
| sum(character_maximum_length)  |
+-----------------------------------------------+
|                                          2771 |
+-----------------------------------------------+
```