**Statistical Monitoring Program Instructions.**

**2. Weekend and national holiday checks**

The function **weekend\_hol\_check** checks whether dates fall on weekends or UK national holidays.

Parameters to give the function:

1. ID and Site:

A data frame containing the participant id (in column 1) and the site name/number (in column 2)

Data frames can be read in with the following code:

**options(stringsAsFactors = FALSE)**

**reg.data<- data.frame(read.table("STUDY12\_REG.txt", row.names=NULL, header=TRUE, sep="\t" ))**

(This would read in a text file called STUDY12\_REG.txt and store it in the data frame reg.data.)

All columns apart from the ID and site should be deleted.

1. Data sheets to check:

A list of data frames to be checked. Each should have the participant ID in the first column followed by all of the dates to be checked. All dates should be in the form dd/mm/yyyy.

Data frames can be read in as in 1 (above) and combined into a list using:

**to.be.checked<- list(reg.data<-“STUDY 12 REG”, treat.data<- “STUDY 12 Treatment”, fup.data<- “STUDY 12 Follow-up data”, death.data<- “STUDY 12 Death”)**

(This would create a list (called “to.be.checked”) of data frames with the data for the registration, treatment, follow-up and death data. Names given to each data frame here will be used in the output files created i.e. the registration data is named “STUDY 12 REG”).

1. Scottish Sites:

3-4 Are two vectors listing the site name/numbers of any sites in Scotland or Northern Ireland (different national holidays). If no Scottish or Northern Irish sites are present these should be set as NULL.

For example if the Scottish sites are numbers 3,5, and 18 then a vector can be created as so:

**scotland.sites<- c(3,5,18)**

If the sites are given as names then the vector can be specified as:

**scotland.sites<- c(“Beatson”, “Ninewells”, “Monklands”)**

1. Northern Irish Sites (as above)
2. The name of the trial.

This will be used to label the output files. For example:

**trial.name<- “STUDY12”**

(Or the trial name can typed as an option when calling the function i.e. “Study12”)

1. Option. This should be set as either "weekends" to check for weekends or "holiday" to check for national holidays.

National holidays the program looks for:

1. New Year’s Day (January 1st or the following Monday if it falls on a weekend)
2. Winter Bank holiday – SCOTLAND ONLY (January 2nd – or the following Monday or Tuesday if it falls on a weekend)
3. St Patrick’s day- NORTHERN IRELAND ONLY (17th of March or the following Monday if it falls on a weekend)
4. Good Friday – the Friday before Easter (March/April - variable dates)
5. Easter Sunday – the first Sunday after the Paschal Moon (the function has a matrix to determine this date depending on the year (will work for the years 1900-2199)
6. Easter Monday – the Monday after Easter Sunday
7. Early May Holiday – The first Monday in May
8. Late May Holiday – The last Monday in May (Moved to the 5th of June in 2012 to create a four day weekend for the Queen’s Jubilee.)
9. Queen’s Jubilee – above and the 4th of June 2012.
10. The Royal Wedding – 29th of April 2011.
11. Orangeman’s Day – NORTHERN IRELAND ONLY (12th of July, or the following Monday if it falls on a weekend.
12. Summer bank holiday – Last Monday in August (England and Northern Ireland) or the first Monday in August (Scotland).
13. St Andrew’s Day – 2nd December SCOTLAND ONLY (or the following Monday if It falls on the weekend)
14. Christmas – 25th December or the following Monday/Tuesday if it falls on the weekend
15. Boxing day – 26th of December or the Following Monday/Tuesday if it falls on weekend )

It should be relatively simple to add new holidays to the code.

A one off holiday (i.e. a holiday on a specific date; for this example the first of November 2013) can be added using the following code:

**newhol<- subset(date, date[,3]=="01/11/2013") ##finds the subset of the data which falls on that date**

**if(dim(newhol)[1]>0){**

**newhol[,9]<-rep("New November holiday", dim(newhol)[1]) ## if any were found they are labelled and**

**holidaystemp<-rbind(holidaystemp, newhol) ## added to a matrix which collects the holidays.**

**}**

If the new holiday is on a specific date every year (1st of November yearly) then then the following code can be used:

**newhol<-subset(date, date[,6]=="01/11")** ## instead of using the whole date this uses just the day and month

**if(dim(newhol)[1]>0){**

**newhol[,9]<-rep("November 1st holiday", dim(newhol)[1]**

**holidaystemp<-rbind(holidaystemp, newhol)**

**}**

If this is the case they we may also want a day in lieu if this falls on a weekend.

##The 2nd or 3rd of November will be the holiday if it is a Monday.

**newhollieu<-subset(date, (date[,6]=="02/11” | date[,6]==”03/11”) & date[,5]==4) ## we use just the day**

**if(dim(newhollieu)[1]>0){ ## and month again but we also specify that if must be a Monday (date[,5]==4 signifies a Monday) newhollieu [,9]<-rep("1st November in Lieu", dim(newhollieu)[1])**

**holidaystemp<-rbind(holidaystemp, newhollieu)**

**}**

If more complicated rules are needed (e.g. first of last Mondays then the existing code should be adjustable)

**Calling the function**

Once the program and the parameters above are stored in R’s memory the program can be run using the following command:

To look for weekends:

**weekend\_hol\_check(id\_and\_sites, to.be.checked, NULL, NULL, “STUDY12”, “weekends”)**

Note we can set the northern irish and Scottish sites as “NULL” as weekends will be the same in all counties.

To look for holidays:

**weekend\_hol\_check(id\_and\_sites, to.be.checked, scotland.sites, n.ireland.sites, “STUDY12”, “holiday”)**

Where each parameter is stored as in 1-6.

**The output:**

The program outputs 2 text files.

The first is called has a name in the form: *“TRIALNAME\_* WEEKEND*\_DATES \_2013-09-02.txt” or “TRIALNAME\_* *HOLIDAY\_DATES \_2013-09-02.txt”*

Where TRIALNAME is the name of the trial and the date would be replaced with the date the program was run.

If “weekends” is specified this will be a text file in the form:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **Site** | **Sheet** | **Variable** | **date** | **Day** |
| 102 | 11 | STUDY12\_REG | DDIAG | 12/10/2003 | Sunday |
| 295 | 60 | STUDY12\_REG | DDIAG | 08/05/2004 | Saturday |
| 323 | 36 | STUDY12\_REG | DDIAG | 25/07/2004 | Sunday |
| 379 | 48 | STUDY12\_REG | DDIAG | 18/09/2004 | Saturday |
| 544 | 83 | STUDY12\_REG | DDIAG | 29/05/2005 | Sunday |
| 678 | 24 | STUDY12\_REG | DDIAG | 27/11/2005 | Sunday |
| 682 | 36 | STUDY12\_REG | DDIAG | 27/11/2005 | Sunday |
| 25 | 36 | STUDY12\_FUP | DFUP | 12/11/2006 | Sunday |
| 45 | 7 | STUDY12\_FUP | DFUP | 09/05/2004 | Sunday |

If “holiday” is specified then the text file will be in the form:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **Site** | **Sheet** | **Variable** | **Date** | **Day** |
| 25 | 168 | RAND | RNDRandDate | 17/03/2011 | St Patrick's Day (Northern Ireland Only) |
| 1 | 317 | RAND | RNDRandDate | 03/05/2010 | Early May Bank Holiday |
| 25 | 172 | RAND | RNDRandDate | 05/06/2012 | Late May Bank Holiday |
| 25 | 169 | RAND | RNDRandDate | 29/08/2011 | Summer Bank Holiday (England and Wales) |
| 34 | 307 | RAND | RNDRandDate | 01/08/2011 | Summer Bank Holiday (Scotland) |
| 1 | 1 | FUP | FUAssessmntDate | 05/04/2010 | Easter Monday |
| 27 | 195 | FUP | FUAssessmntDate | 01/08/2011 | Summer Bank Holiday (Scotland) |
| 24 | 167 | FUP | FUTuRespAssDate | 05/04/2010 | Easter Monday |
| 30 | 252 | FUP | FUTuRespAssDate | 05/06/2012 | Late May Bank Holiday |
| 11 | 160 | treat\_assess\_1month | AssessDate | 01/01/2008 | New Year's Day |
| 11 | 270 | treat\_assess\_1month | AssessDate | 25/04/2011 | Easter Monday |
| 25 | 173 | treat\_assess\_1month | AssessDate | 02/04/2010 | Good Friday |
| 27 | 195 | treat\_assess\_1month | AssessDate | 04/06/2012 | Queen's Jubilee 2012 |

Note that the example above was not real data but was a dataset to which national holiday, had been added.

If no weekends or holidays are found the file will contain the message "No weekend dates found" or "No holiday dates found".

The second text file has a name in the form: *“TRIALNAME\_vars\_checked\_weekends\_2013-09-02” or “TRIALNAME\_vars\_checked\_holidays\_2013-09-02”* and contains information about the numbers of sheets, variables, values etc. checked:

**Warnings:**

There are no error messages coded into the function. If data is not read in as above it may not work as the function should or possibly at all. Please take care when creating the parameters from your data.

Please ensure that the dates you are checking should be checked. For example, a date of randomisation should not fall on a weekend or bank holiday (if patients are randomised by a trials centre) but a date of death could fall on any day.