Requirements:

Skip first 11 lines

Skip any line starting with tab

For remaining lines, use custom regex.

# Step-by-step preprocess

## Open file

#Replace file, xmlFile as required

file <- "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-2\\1-Data||0-raw\\OPERPAR1\_SystemOut\_13.05.20\_09.04.50.log"

library(gsubfn)

#Switch lines to read full log or summary

#lines <- readLines(file, 20)

lines <- readLines(file)

## Skip header lines, other lines

#Identify starts of headers

startEnvIndex <- grep(" Start Display Current Environment ",lines)

#Each header is 11 lines long

#Get index of all header lines

myF <- function(x)

{x:(x+10)}

envIndex <- c(sapply(startEnvIndex,myF, simplify="array"))

#Skip the following types of lines

# - Lines starting with tab

# - Lines starting with space

# - Header lines

notread <- c(grep("^\t",lines),grep("^ ",lines), envIndex)

lines2 <- lines[setdiff(1:length(lines),notread)]

## Create data frame

# Regex for this table

#regex <- "(.{28}) (.{8}) (\S\*)\s\*(\w)\s\*(.\*)\r"

regex <- "(.{28}) (.{8}) ([\\S\*)\\s\*(\\w)\\s\*(.\*)](file:///\\S*)\\s*(\\w)\\s*(.*))"

#Create data frame, apply regex

log<- data.frame(strapplyc(lines2, regex, simplify = "rbind"))

names(log) <- c("date","code","source","severity","message")

## Handle log severity levels

Initially, the log table has the following values for the severity column:

A E I O W

350 3066 10515 1423 4630

~~We interpret these as follows:~~

* ~~A: Alert~~
* ~~E: Error~~
* ~~I: Info~~
* ~~O: Notice~~
* ~~W: Warning~~
* A: Audit - 2
* E:Error - 4
* I: Info- 1
* O: System.Out - 0
* W: Warning - 3

#levels(factors) <- c("audit","error", "info", "sysOut", "warning")

levels(log$severity) <- c("2","4", "1", "0", "3")

log$sevNum <- as.numeric(as.character(log$severity))

## Handle Dates

date2 <- gsub("(.{18}):","\\1.", log$date)

date3 <- strptime(date2, "[%d/%m/%y %H:%M:%OS")

log$dateNum <- date3

log <- log[c("dateNum", "code", "source", "sevNum", "message")]

#plot(log2dateNum,log$sevNum)

# Preprocess R code

#Replace file, xmlFile as required

file <- "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-2\\OPERPAR1\_SystemOut\_13.05.20\_09.04.50.log"

library(gsubfn)

#Switch lines to read full log or summary

#lines <- readLines(file, 20)

lines <- readLines(file)

#Identify starts of headers

startEnvIndex <- grep(" Start Display Current Environment ",lines)

#Each header is 11 lines long

#Get index of all header lines

myF <- function(x)

{x:(x+10)}

envIndex <- c(sapply(startEnvIndex,myF, simplify="array"))

#Skip the following types of lines

# - Lines starting with tab

# - Lines starting with space

# - Header lines

notread <- c(grep("^\t",lines),grep("^ ",lines), envIndex)

lines2 <- lines[setdiff(1:length(lines),notread)]

# Regex for this table

#regex <- "(.{28}) (.{8}) (\S\*)\s\*(\w)\s\*(.\*)\r"

regex <- "(.{28}) (.{8}) ([\\S\*)\\s\*(\\w)\\s\*(.\*)](file:///\\S*)\\s*(\\w)\\s*(.*))"

#Create data frame, apply regex

log<- data.frame(strapplyc(lines2, regex, simplify = "rbind"))

names(log) <- c("date","code","source","severity","message")

#levels(factors) <- c("audit","error", "info", "sysOut", "warning")

levels(log$severity) <- c("2","4", "1", "0", "3")

log$sevNum <- as.numeric(as.character(log$severity))

date2 <- gsub("(.{18}):","\\1.", log$date)

date3 <- strptime(date2, "[%d/%m/%y %H:%M:%OS")

log$dateNum <- date3

log <- log[c("dateNum", "code", "source", "sevNum", "message")]

#plot(log2dateNum,log$sevNum)

## Initial table

dateNum code source sevNum message

1 2013-05-16 21:53:49 0000001c EJBContainerI 1 (...)

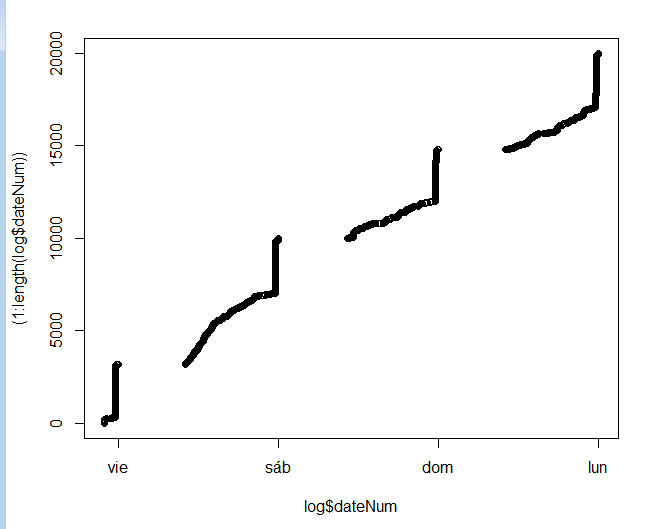
# Exploration

Due to the number of log lines, we will often use a log sample during exploration

minilog<- log[sort(sample(1:length(log$dateNum), size=300, replace=FALSE)),]

## Daily Event patterns

The application servers observed in these logs are shut down every night, resulting in huge spikes of events:



par(mfrow=c(2,2))

par(pch=1)

par(cex=0.5)

minilogIndex <- minilog$dateNum$hour==23 & (minilog$dateNum$min>=30 & minilog$dateNum$min<= 40)

logIndex <- log$dateNum$hour==23 & (log$dateNum$min>=30 & log$dateNum$min<= 40)

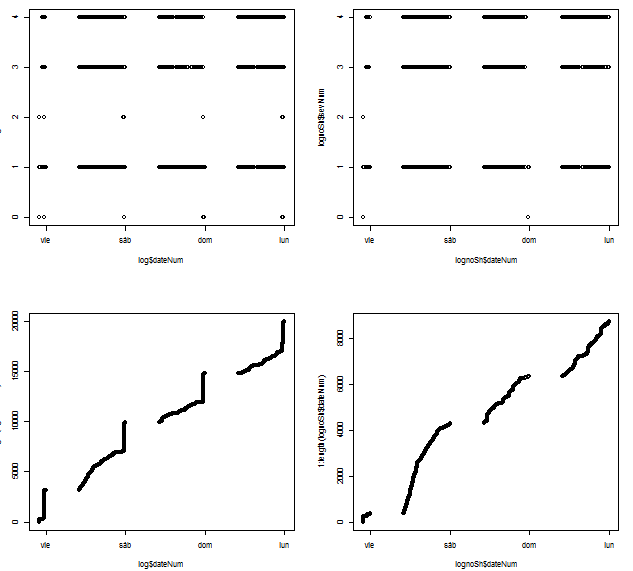
lognoSh <- log[!logIndex,]

plot(log$dateNum,log$sevNum)

plot(lognoSh$dateNum,lognoSh$sevNum)

plot(log$dateNum,1:length(log$dateNum))

plot(lognoSh$dateNum,1:length(lognoSh$dateNum))



### tmp

> plot(minilog$dateNum,(1:300), pch=1, cex=0.5, col=rgb(0,0,0,0.5))

> plot(minilog$dateNum,(1:300), pch=1, cex=0.5, col=rgb(0,0,0,0.5))

> plot(minilog$dateNum[minilogIndex],(1:300)[minilogIndex], pch=1, cex=0.5, col=rgb(0,0,0,0.5))

> plot(minilog$dateNum[!minilogIndex],(1:300)[!minilogIndex], pch=1, cex=0.5, col=rgb(0,0,0,0.5))

table(log$dateNum$wday)

0 4 5 6

5197 3197 6779 4811

logIndex1 <- log$dateNum$hour==23 & (log$dateNum$min>=30 & log$dateNum$min<= 40) & log$dateNum$wday==4

logIndex2 <- log$dateNum$hour==23 & (log$dateNum$min>=30 & log$dateNum$min<= 40) & log$dateNum$wday==5

logIndex3 <- log$dateNum$hour==23 & (log$dateNum$min>=30 & log$dateNum$min<= 40) & log$dateNum$wday==6

logIndex4 <- log$dateNum$hour==23 & (log$dateNum$min>=30 & log$dateNum$min<= 40) & log$dateNum$wday==0

log[logIndex1,][1,]

date code source severity

333 [16/05/13 23:30:47:904 CEST] 0000030c AdminHelper 5

message

333 ADMN1020I: Se ha intentado detener el servidor OPERPAR1\_1. (ID de usuario = defaultWIMFileBasedRealm/aplwasadm)

sevNum dateNum

333 5 2013-05-16 23:30:47

> log[logIndex2,][1,]

date code source severity

7020 [17/05/13 23:30:50:829 CEST] 000001fe AdminHelper 5

message

7020 ADMN1020I: Se ha intentado detener el servidor OPERPAR1\_1. (ID de usuario = defaultWIMFileBasedRealm/aplwasadm)

sevNum dateNum

7020 5 2013-05-17 23:30:50

> log[logIndex3,][1,]

date code source severity

11971 [18/05/13 23:30:50:859 CEST] 0000015e AdminHelper 5

message

11971 ADMN1020I: Se ha intentado detener el servidor OPERPAR1\_1. (ID de usuario = defaultWIMFileBasedRealm/aplwasadm)

sevNum dateNum

11971 5 2013-05-18 23:30:50

> log[logIndex4,][1,]

date code source severity

17051 [19/05/13 23:30:54:732 CEST] 00000143 AdminHelper 5

message

17051 ADMN1020I: Se ha intentado detener el servidor OPERPAR1\_1. (ID de usuario = defaultWIMFileBasedRealm/aplwasadm)

sevNum dateNum

17051 5 2013-05-19 23:30:54

### Tmp: Searching for specific time, date

How to identify the inflection point? At which time is the server shut down?

<http://stackoverflow.com/questions/8126537/extract-date-elements-from-posixlt-and-put-into-data-frame-in-r>

POSIXlt objects are a list of 9 components (see the Details section of ?POSIXlt for more information). Because the dd\_mmm\_yy column is POSIXlt, you don't need a function to extract the components. You can just extract the components by their names:

orders$day <- orders$dd\_mmm\_yy$mday # day of month

orders$month <- orders$dd\_mmm\_yy$mon+1 # month of year (zero-indexed)

orders$year <- orders$dd\_mmm\_yy$year+1900 # years since 1900

orders

# order\_id dd\_mmm\_yy day month year

# 1 1 2005-07-28 28 7 2005

# 2 2 2007-03-04 4 3 2007

<http://127.0.0.1:15212/library/base/html/DateTimeClasses.html>

There are two basic classes of date/times. Class "POSIXct" represents the (signed) number of seconds since the beginning of 1970 (in the UTC timezone) as a numeric vector. Class "POSIXlt" is a named list of vectors representing:

* sec
* min
* hour
* mday
* mon
* year
* wday
* yday
* isdst

Example:

minilogIndex <- minilog$dateNum$hour==23 & (minilog$dateNum$min>30 & minilog$dateNum$min< 40)

Classes "POSIXct" and "POSIXlt" are able to express fractions of a second. (Conversion of fractions between the two forms may not be exact, but will have better than microsecond accuracy.)

<http://127.0.0.1:15212/library/base/html/round.POSIXt.html>

Round date-time objects

round(x,"mins")

## Checking log levels

### New information

Logs come from a WebSphere Application Server

<http://publib.boulder.ibm.com/infocenter/wsdoc400/v6r0/index.jsp?topic=/com.ibm.websphere.iseries.doc/info/ae/ae/ctrb_enabletrc.html>

Both Java logging levels and WebSphere Application Server levels can be used. The following is a complete list of valid levels, ordered in ascending order of severity:

1. all
2. finest or debug
3. finer or entryExit
4. fine or event
5. detail
6. config
7. **info**
8. **audit**
9. **warning**
10. severe or **error**
11. fatal
12. off

<http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?topic=%2Frzamy%2F50%2Ftrb%2Ftrbjvmintro.htm>

EventType

A one character field that indicates the type of the message or trace event. Message types are in upper case. Possible values include:

A - An audit message.

I - An informational message.

W - A Warning message.

E - An Error message.

F - A Fatal message.

O - A message that was written directly to System.out by the user application or application server internal components.

R - A message that was written directly to System.err by the user application or application server internal components.

u - A specialized message type used by the message logging component of the application server runtime.

Z - A placeholder to indicate the type was not recognized.

### Exploration analysis

Let's filter the log and keep only level 3-4 events (warnings and errors)

1. all level 3-4 events
2. level 3-4 events minus shutdown periods

log34 <- log[log$sevNum>=3,]

lognoSh34 <- lognoSh[lognoSh$sevNum>=3,]

plot(log34 $dateNum,1:length(log34 $dateNum))

plot(lognoSh34 $dateNum,1:length(lognoSh34 $dateNum))

lognoSh3 <- lognoSh34[lognoSh34$sevNum==3,]

plot(lognoSh3 $dateNum,1:length(lognoSh3 $dateNum))

lognoSh4 <- lognoSh34[lognoSh34$sevNum==4,]

plot(lognoSh4 $dateNum,1:length(lognoSh4 $dateNum))

write.csv(lognoSh3, "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-2\\tmp\\log3.txt")

write.csv(lognoSh4, "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-2\\tmp\\log4.txt")

, the log table has the following values for the eventType column:

A E I O W

350 3066 10515 1423 4630

* A: Audit - 2
* E:Error - 4
* I: Info- 1
* O: System.Out - 0
* W: Warning - 3

2 4 1 0 3

350 3066 10515 1423 4630

0 1 2 3 4

1423 10515 350 4630 3066

### Initial Assumptions

Initially, the log table has the following values for the severity column:

A E I O W

350 3066 10515 1423 4630

We interpret these as follows:

* A: Alert - Syslog 1 -> 5
* E: Error - Syslog 3 -> 4
* I: Info - Syslog 6 -> 1
* O: Notice - Syslog 5 -> 2
* W: Warning - Syslog 4 ->3

<http://en.wikipedia.org/wiki/Syslog#Severity_levels>

table(log$sevNum)

1 2 3 4 5

10515 1423 4630 3066 350

Does this severity match actual messages? Example log lines:

table(minilog$sevNum)

1 2 3 4 5

166 27 61 43 3

### Level 5 (¿Alert?)

#ALERT?

> head(minilog$message[minilog$sevNum==5])

[1] WTRN0133I: Se ha completado el proceso de recuperación de transacción para este servidor

[2] WSVR0217I: Se está deteniendo la aplicación: CN\_Cuentas\_SANINT\_ENS

[3] WSVR0204I: Aplicación: JMX\_\_OPERPAR1\_1\_war Nivel de build de aplicación: Desconocido

5794 Levels: ... WTRN0135I: El servicio de transacción no está recuperando transacciones.

> table(factor(log$source[log$sevNum==5]))

AdminHelper AdminInitiali ApplicationMg CompositionUn Configuration CSIClientRI CSIServerRI

4 4 141 57 4 4 4

FailureScopeC JMSRegistrati JMXSoapAdapte NameServerImp PMIImpl RecoveryManag RMIConnectorC

4 4 4 4 4 16 4

SecurityCompo SecurityConne ServerCollabo ServiceInit UserRegistryI WorkAreaServi WSChannelFram

4 4 4 12 4 4 56

WsServerImpl

4

A note on log sources: Any source with exactly 4 messages can be discarded, since those will most likely be due to the (4) daily shutdown/restart cycles present in the log.

Likewise, sources with (multiple of 4) messages are suspect for discarding

### Level 4 (¿Error?)

#ERROR

> head(minilog$message[minilog$sevNum==4])

[1] WLTC0017E: Los recursos se han retrotraído porque se ha llamado a setRollbackOnly().

[2] org.apache.fop.fo.flow.ExternalGraphic bind Image not available:

[3] org.apache.fop.fo.flow.ExternalGraphic bind Image not available:

[4] org.apache.fop.fo.flow.ExternalGraphic bind Image not available:

[5] WLTC0017E: Los recursos se han retrotraído porque se ha llamado a setRollbackOnly().

[6] com.ibm.ws.webcontainer.servlet.ServletWrapper service SRVE0068E: Se ha creado una excepción no detectada en uno de los métodos de servicio del servlet /CN\_Recibos\_Pres/verReciboPDF\_CNRec\_V2/verRecibopdf.jsp en la aplicación CN\_Recibos\_SANINT\_ENS. Excepción creada: java.lang.IllegalStateException: SRVE0199E: Ya se ha obtenido OutputStream

5794 Levels: ... WTRN0135I: El servicio de transacción no está recuperando transacciones.

table(factor(log$source[log$sevNum==4]))

FONode FopImage LocalTranCoor ModuleManifes servlet srt WASSession

1280 640 556 4 572 4 6

WASSessionCor webcontainer

3 1

### Level 3 (¿Warning?)

#WARNING

> head(minilog$message[minilog$sevNum==3])

[1] UTLS0005W: Los atributos MANIFEST de un paquete opcional instalado en la biblioteca compartida ARQ\_RIGEL\_V35SP4F00 están en conflicto y alterarán los incluidos en los atributos MANIFEST de la biblioteca compartida ARQ\_RIGEL\_V34SP3F00.

[2] UTLS0005W: Los atributos MANIFEST de un paquete opcional instalado en la biblioteca compartida ARQ\_RIGEL\_V35SP3F03 están en conflicto y alterarán los incluidos en los atributos MANIFEST de la biblioteca compartida ARQ\_RIGEL\_V35SP1F03.

[3] UTLS0005W: Los atributos MANIFEST de un paquete opcional instalado en la biblioteca compartida ARQ\_RIGEL\_V35SP1F03 están en conflicto y alterarán los incluidos en los atributos MANIFEST de la biblioteca compartida ARQ\_RIGEL\_V35SP0F02.

[4] org.apache.fop.layoutmgr.table.TableContentLayoutManager createElementsForRowGroup The contents of row 18 are taller than they should be (there is a block-progression-dimension or height constraint on the indicated row). Due to its contents the row grows to 29520 millipoints, but the row shouldn't get any taller than MinOptMax[min=opt=max=20000] millipoints. (fo:table-row, "01/04/2013,  , 01/04/2013,  , PAGO RECIBO DE SANTANDER CONSUMER F)

[5] org.apache.fop.image.analyser.ImageReaderFactory make No ImageReader found for wsjar:file:/D:/WebSphere7/AppServer/profiles/Node01/installedApps/PARTICULARES\_CELDA2\_REAL/CN\_Cuentas\_SANINT\_ENS.ear/scenario.jar!/CN\_Cuentas\_LN/

[6] org.apache.fop.layoutmgr.table.TableContentLayoutManager createElementsForRowGroup The contents of row 192 are taller than they should be (there is a block-progression-dimension or height constraint on the indicated row). Due to its contents the row grows to 29520 millipoints, but the row shouldn't get any taller than MinOptMax[min=opt=max=20000] millipoints. (fo:table-row, "03/12/2012,  , 03/12/2012,  , PAGO RECIBO DE SECURITAS DIRECT ESP)

5794 Levels: ... WTRN0135I: El servicio de transacción no está recuperando transacciones.

table(factor(log$source[log$sevNum==3]))

BreakingAlgor FfdcProvider ImageReaderFa jar ModuleManifes TableContentL ThreadPoolMgr

33 23 640 28 676 3214 4

WSDynamicPoli WSKeyStore

8 4

### Level 2 (System Out)

#SYS.OUT

> head(minilog$message[minilog$sevNum==2])

[1] 16/05 21:53:54.262 (I) [Default : 5 ] 'kernel': Creando el bootstrap CfgManager para el kernel 'Kernel\_CN\_Cuentas\_SANINT\_ENS\_V1'

[2] 16/05 23:30:57.202 (I) [rable Alarm : 3] 'kernel': El kernel 'Kernel\_JMX\_OPERPAR1\_1' ha sido parado correctamente

[3] 16/05 23:34:35.499 (D) [\_JMX\_OPERPAR1\_1] 'kernel.bootCfgMngr': Resolviendo url: 'local:#/configuration/configurationBuilders/encryptedDBPropertiesMimeType'

[4] 16/05 23:34:44.132 (I) [ver.startup : 2] 'kernel': Se han fusionado los ficheros de configuracion de servicos para el kernel 'Kernel\_CN\_Depositos\_SANINT\_ENS\_V1'

[5] 16/05 23:34:48.374 (W) [ver.startup : 2] 'kernel.logManager': [BKSW01005000002] : La categoría 'root.dynamicLoader' no ha sido declarada, usado el logger trastero en su lugar

[6] 16/05 23:34:50.527 (I) [ver.startup : 0] 'kernel': Inicializando el kernel 'Kernel\_CN\_Transferencias\_SANINT\_ENS\_V1'

5794 Levels: ... WTRN0135I: El servicio de transacción no está recuperando transacciones.

table(factor(log$source[log$sevNum==2]))

SystemOut

1423

#INFO

> head(minilog$message[minilog$sevNum==1])

[1] WSVR0041I: Se está deteniendo el jar de EJB: BDPGPO\_MantenerRelContrato\_ESP\_LN.jar

[2] com.ibm.ws.webcontainer.servlet.ServletWrapper doDestroy SRVE0253I: [CN\_Transferencias\_SANINT\_ENS] [/CN\_Transferencias\_SANINT\_ENS] [/CN\_Transferencias\_Pres/ListarTransfHabitual\_CNTransf\_V1/DatosTransfHabituales.jsp]: Destrucción satisfactoria.

[3] WSVR0041I: Se está deteniendo el jar de EJB: TransferCobros\_LN.jar

[4] WSVR0041I: Se está deteniendo el jar de EJB: FACTEC\_UserManagementER\_LN.jar

[5] WSVR0059I: Se ha detenido el jar de EJB: BDPABB\_DatosEconomicos\_LN.jar

[6] WSVR0041I: Se está deteniendo el jar de EJB: REGSEG\_CcggSerDirInter\_LN.jar

5794 Levels: ... WTRN0135I: El servicio de transacción no está recuperando transacciones.

>

table(factor(log$source[log$sevNum==1]))

ActivitySessi ApplicationRe AppProfileCom authz CacheServiceI CGBridgeServi CGBridgeSubsc

8 4 8 4 4 4 8

collaborator ComponentMeta Config CoordinatorIm CScopeCompone DCSPluginSing DefaultApplic

4 4 8 4 4 4 4

DiagnosticCon DiscoveryMBea distSecurityC EJBContainerI FFDCJanitor I18nService I18nServiceSe

4 4 28 7186 5 4 4

InternalDB2Un InternalGener IPCConnectorS J2EEServiceMa JAASLoginConf JMSRegistrati JMXConnectors

28 112 4 8 4 4 4

JmxConsoleAct JPAComponentI ManagerAdmin ModelMgr ObjectCacheSe ObjectPoolSer PluginConfigS

8 20 8 8 4 8 4

PropertyMessa ProviderTrack RecoveryDirec RegistryCache RegistryConfi ResourceMgrIm SchedulerServ

12 8 8 4 4 96 32

SecurityDM ServerCache servlet SibMessage SOAPAcceptorC SOAPContainer SSLComponentI

4 32 1162 8 4 4 16

SSLConfigMana SSLDiagnostic StartUpServic TableLayoutMa TCPChannel UserManagemen WASAxis2Compo

4 4 4 1280 40 16 4

WASSessionCor webapp webcontainer WebsphereComm WSJdbcConnect WSRdbDataSour

29 71 105 4 1 28

Level 2 messages (originally labelled as severity=="O") always correspond to System.Out messages.

* They have their own severity levels!

sysOutMessages <- log$message[log$sevNum==2]

write(sysOutMessages, "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-2\\sysOutMessages.txt")

Initial impression: Sys.Out logs can be safely ignored!

log2 <- log[log$sevNum>=3,]

log3 <- log2[,c(7,6,3,5)]

log3$source <- factor(log3$source)

log3$message <- factor(log3$message)

plot(log3$dateNum,1:length(log3$dateNum), pch=1, cex=0.5, col=rgb(0,0,0,0.5))

log4 <- log[log$sevNum<3,]

log5 <- log4[,c(7,6,3,5)]

log5$source <- factor(log5$source)

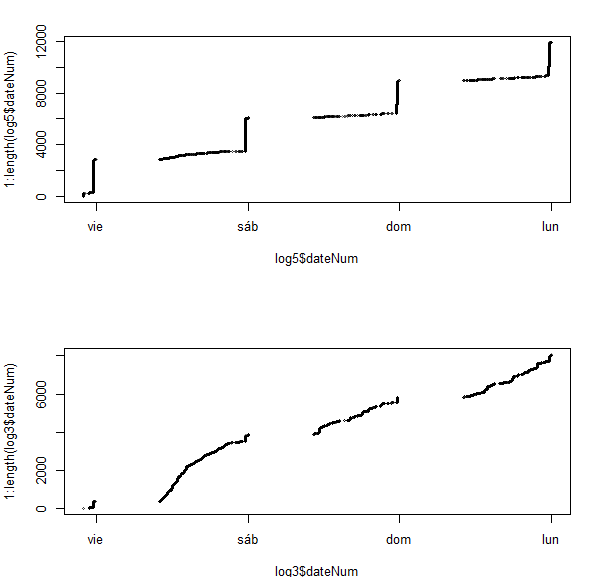
log5$message <- factor(log5$message)

plot(log5$dateNum,1:length(log5$dateNum), pch=1, cex=0.5, col=rgb(0,0,0,0.5))

### Removing log levels 1,2

After removing logs with level 1 (info) and 2 (sys.out), the huge spikes at shutdown time mostly disappear: (log5 has levels 1-2, log3 has levels 3-5)

plot(log5$dateNum,1:length(log5), pch=1, cex=0.5, col=rgb(0,0,0,0.5))



write.csv(log3, "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-2\\logSev345.csv")

write.table(log5, "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-2\\logSev12.csv")

## Checking log sources

1. Take table of sources per log level
2. Diff tables to identify
   1. sources that appear at multiple severity levels
   2. sources that appear at single severity level
3. Identify sources providing relevant information
   1. Filter sources exclusively related to shutdown messages

### Source tables

> table(factor(log$source[log$sevNum==5]))

AdminHelper AdminInitiali ApplicationMg CompositionUn Configuration CSIClientRI CSIServerRI

4 4 141 57 4 4 4

FailureScopeC JMSRegistrati JMXSoapAdapte NameServerImp PMIImpl RecoveryManag RMIConnectorC

4 4 4 4 4 16 4

SecurityCompo SecurityConne ServerCollabo ServiceInit UserRegistryI WorkAreaServi WSChannelFram

4 4 4 12 4 4 56

WsServerImpl

4

table(factor(log$source[log$sevNum==4]))

FONode FopImage LocalTranCoor ModuleManifes servlet srt WASSession

1280 640 556 4 572 4 6

WASSessionCor webcontainer

3 1

table(factor(log$source[log$sevNum==3]))

BreakingAlgor FfdcProvider ImageReaderFa jar ModuleManifes TableContentL ThreadPoolMgr

33 23 640 28 676 3214 4

WSDynamicPoli WSKeyStore

8 4

table(factor(log$source[log$sevNum==1]))

ActivitySessi ApplicationRe AppProfileCom authz CacheServiceI CGBridgeServi CGBridgeSubsc

8 4 8 4 4 4 8

collaborator ComponentMeta Config CoordinatorIm CScopeCompone DCSPluginSing DefaultApplic

4 4 8 4 4 4 4

DiagnosticCon DiscoveryMBea distSecurityC EJBContainerI FFDCJanitor I18nService I18nServiceSe

4 4 28 7186 5 4 4

InternalDB2Un InternalGener IPCConnectorS J2EEServiceMa JAASLoginConf JMSRegistrati JMXConnectors

28 112 4 8 4 4 4

JmxConsoleAct JPAComponentI ManagerAdmin ModelMgr ObjectCacheSe ObjectPoolSer PluginConfigS

8 20 8 8 4 8 4

PropertyMessa ProviderTrack RecoveryDirec RegistryCache RegistryConfi ResourceMgrIm SchedulerServ

12 8 8 4 4 96 32

SecurityDM ServerCache servlet SibMessage SOAPAcceptorC SOAPContainer SSLComponentI

4 32 1162 8 4 4 16

SSLConfigMana SSLDiagnostic StartUpServic TableLayoutMa TCPChannel UserManagemen WASAxis2Compo

4 4 4 1280 40 16 4

WASSessionCor webapp webcontainer WebsphereComm WSJdbcConnect WSRdbDataSour

29 71 105 4 1 28

### Intersections

intersect(factor(log$source[log$sevNum==1]),factor(log$source[log$sevNum==5]))

[1] "JMSRegistrati"

4/4

intersect(factor(log$source[log$sevNum==1]),factor(log$source[log$sevNum==4]))

[1] "servlet" "WASSessionCor" "webcontainer"

1162/572 29/3 105/1

intersect(factor(log$source[log$sevNum==3]),factor(log$source[log$sevNum==4]))

[1] "ModuleManifes"

676/4

### Source Tables after fitering shutdown

logShutdownIndex3 <- log3$dateNum$hour==23 & (log3$dateNum$min>=30 & log3$dateNum$min<= 59)

log345noSh <- log3[!logShutdownIndex3,]

dim(log345noSh)

6780 4

logShutdownIndex5 <- log5$dateNum$hour==23 & (log5$dateNum$min>=30 & log5$dateNum$min<= 59)

log12noSh <- log5[!logShutdownIndex5,]

dim(log12noSh)

6780 4

write.csv(log345noSh, "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-2\\log345noSh.csv")

write.csv(log12noSh, "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-2\\log12noSh.csv")

# Experiments

## Factor experiments

factors <- tmp$X4

factors

unclass(factors)

f2 <- factor(unclass(factors))

levels(f2)

as.numeric(as.character(sev1))

## date experiments

as.character(log$date[1:20])

as.Date(tmp,format="[%d/%m/%y")

tmp <- as.character(log$date[1])

strftime(tmp,format="[%d/%m/%y)

as.Date(["16/05/13",format="[%d/%m/%y")

strptime("[16/05/13 21:53:49.255 CEST]", "[%d/%m/%y %H:%M:%OS")

NOTE: Date milliseconds can't be parsed due to format: 49:255 should be 49.255

* tgsub("(.{18}):","\\1.", tmp)
* tgsub("(.{18}):","\\1.", tmp)
* tgsub("(.{18}):","\\1.", log$date)

date2 <- gsub("(.{18}):","\\1.", log$date)

date3 <- strptime(date2, "[%d/%m/%y %H:%M:%OS")

log$dateNum <- date3

log2 <- log[200:400,]

> plot(log2$dateNum,log2$sevNum)