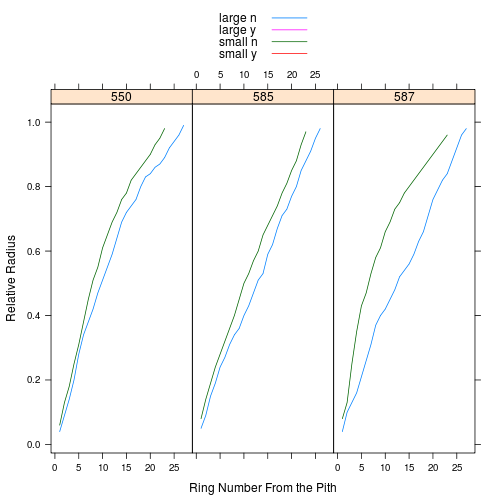
# LS.1.5 Report - Extras

## 1a. Plot of 3 or 5 archetypal growth curves.

library(lattice)  
library(RODBC)  
ch = odbcConnect("LS15")  
R = sqlQuery(ch, "\n select \n LogEndImages.filename,\n Forest,\n logs.ScionLogNumber,\n LogEndImages.logEnd,\n inWhorl,\n ring, \n r\n from \n logs, \n LogEndRings, \n LogEndImages\n where \n LogEndRings.filename=LogEndImages.filename \n and LogEndImages.ScionLogNumber=logs.ScionLogNumber\n ")  
xyplot(r ~ ring | as.factor(ScionLogNumber), R, group = paste(logEnd, inWhorl),   
 type = "l", auto.key = list(points = FALSE, lines = TRUE), xlab = "Ring Number From the Pith",   
 ylab = "Relative Radius", subset = inWhorl == "n" & ScionLogNumber %in%   
 c(585, 550, 587), layout = c(3, 1, 1))



plot of chunk unnamed-chunk-1

load("~/Desktop/Dropbox/LS15/log-v-lumber/logs.RData")

Ring positions were manually digitzed in the yard log end imagery at a single point on the circumference. Limitations imposed by the nature of the log end surfaces, the presence of knots and the quality of the imagery introduce a degree of noise, and on occasion incredible results (e.g. more rings at the small end than at the large). The figure above plots relative growth trajectories for three typical logs. Growth trajectories at both log ends are shown, with both ends free of knots. For the most part the large and small ends show similar patterns. Classes of growth trajectory can be identified:

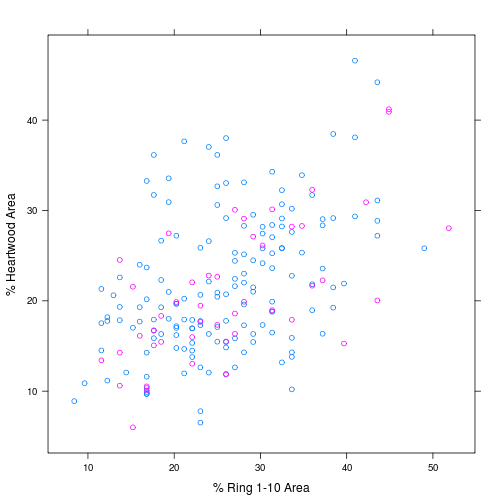
1. Growth rate constant through time at both ends (e.g. 585)
2. Growth rate decreasing through time at both ends (e.g. 550)
3. Growth trajectories differing between ends (e,g, 587)

## 1b. Is ring10volfrac equal to heartVolFrac?

The next plot compares heartwood area fraction with the area fraction occupied by material from pith to ring 10 for each log end. The datapoints are colored according to whether or not the log end contains knots. While the two measures are significantly (but not strongly) correlated, they are not synonymous. It is interesting to note that log ends containing knots, and consequently for which the boundary of ring 10 from the pith is ususally very disturbed, do not stand out.

ch = odbcConnect("LS15")  
E = sqlQuery(ch, "\n select \n LogEndDigitizations.\*,\n LogEndImages.filename,\n Forest,\n logs.ScionLogNumber,\n LogEndImages.logEnd,\n inWhorl,\n Aheart\_mm2/A\_mm2 as heartAreaFrac\n from \n logs, \n LogEndDigitizations, \n LogEndImages\n where \n LogEndDigitizations.filename=LogEndImages.filename \n and LogEndImages.ScionLogNumber=logs.ScionLogNumber\n and LogEndImages.imageType='yard'\n and mm\_per\_pixel is not null")  
  
xyplot(heartAreaFrac \* 100 ~ ring10AreaFrac \* 100, group = inWhorl, E, ylab = "% Heartwood Area",   
 xlab = "% Ring 1-10 Area")

## Warning: closing unused RODBC handle 1



plot of chunk unnamed-chunk-2

summary(lm(heartAreaFrac ~ ring10AreaFrac, E))

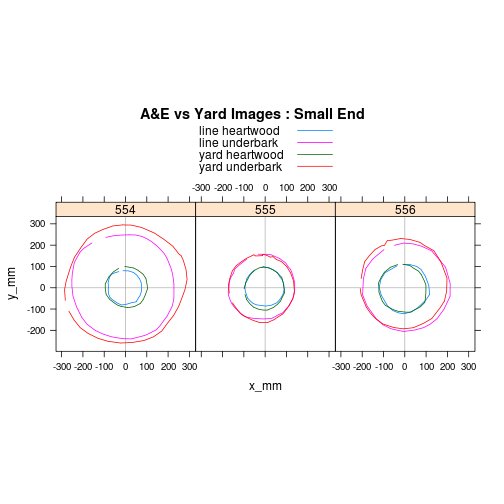
##   
## Call:  
## lm(formula = heartAreaFrac ~ ring10AreaFrac, data = E)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.1477 -0.0491 -0.0050 0.0381 0.1841   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.1028 0.0153 6.73 1.8e-10 \*\*\*  
## ring10AreaFrac 0.4367 0.0557 7.83 2.8e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0675 on 198 degrees of freedom  
## Multiple R-squared: 0.237, Adjusted R-squared: 0.233   
## F-statistic: 61.4 on 1 and 198 DF, p-value: 2.81e-13

## 2. A&E vs yard digitizations to demonstrate that pick out = dryness (554-556). Some words to the same effect.

ch = odbcConnect("LS15")  
P = sqlQuery(ch, "select \* from LogEndDigitizations as d, LogEndDigitizationEdges as e, LogEndDigitizationPoints as p where d.id=e.digitizationID and e.id=p.edgeID and type<>'pith'")

## Warning: closing unused RODBC handle 2

myxyplot = function(x, y, ...) {  
 panel.abline(h = 0, col = "grey70")  
 panel.abline(v = 0, col = "grey70")  
 panel.xyplot(x, y, ...)  
}  
xyplot(y\_mm ~ x\_mm | as.factor(ScionLogNumber), group = paste(imageType, type),   
 P, auto.key = list(lines = TRUE, points = FALSE), type = "l", subset = logEnd ==   
 "small" & ScionLogNumber %in% c(554, 555, 556), layout = c(3, 1, 1),   
 aspect = "iso", panel = myxyplot, main = "A&E vs Yard Images : Small End")



plot of chunk unnamed-chunk-3

The figure above compares, for three typical log small ends, the boundary of the heartwood zone identified in the yard log end imagery with the boundary of the pick-out' zone seen in the A&E images. While the match is not perfect, there is substantial correspondence between the heart/sap boundary, suggesting that 'pick-out' happens when lumen water has been lost, perhaps due to changes in wood-saw friction.

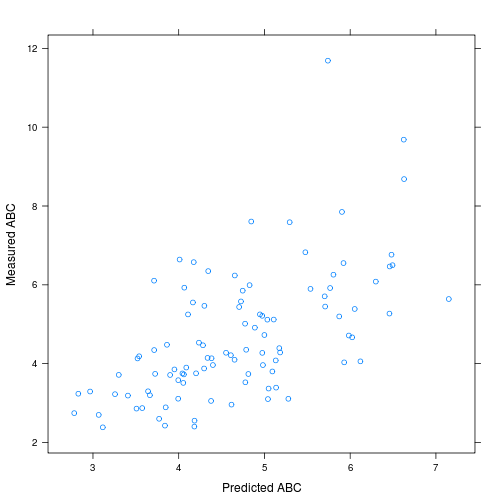
## 3. Meas vs predicted of a linear model for abc(swv, log dia, heart%).

The figure below compares measured ABC with that predicted from a linear model based on measures of log SWV, SED and the fraction of the total volume occupied by material from the first 10 growth rings. While is very poor, the model suggests that these measures could be used to increase the proprtion of good log entering a process.

m.abc = lm(abc.bavg ~ swv.calibre + SED + heartVolFrac + ring10VolFrac, L)  
summary(m.abc)

##   
## Call:  
## lm(formula = abc.bavg ~ swv.calibre + SED + heartVolFrac + ring10VolFrac,   
## data = L)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.171 -0.743 -0.204 0.492 5.947   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 25.893558 3.436082 7.54 2.8e-11 \*\*\*  
## swv.calibre -0.005417 0.000805 -6.73 1.3e-09 \*\*\*  
## SED -0.012198 0.003078 -3.96 0.00014 \*\*\*  
## heartVolFrac 6.239887 2.241777 2.78 0.00649 \*\*   
## ring10VolFrac -1.136503 2.916566 -0.39 0.69765   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.32 on 95 degrees of freedom  
## Multiple R-squared: 0.36, Adjusted R-squared: 0.334   
## F-statistic: 13.4 on 4 and 95 DF, p-value: 1.09e-08

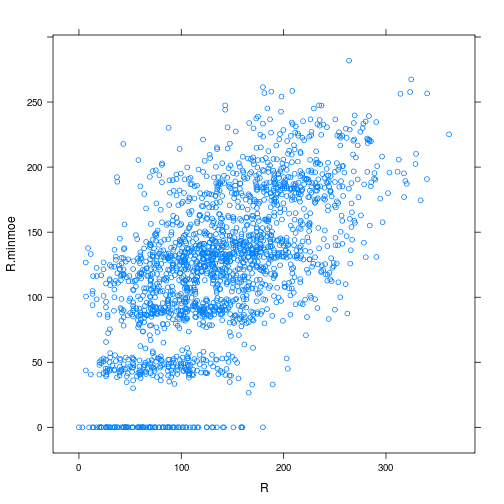
xyplot(L$abc.bavg ~ predict(m.abc), xlab = "Predicted ABC", ylab = "Measured ABC")



plot of chunk unnamed-chunk-4

## 4. recompute R based on centre of low stiffness boards (c.f. pith)

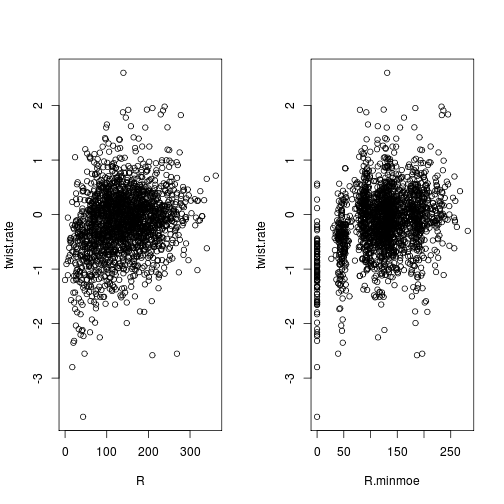
# in each log find the least stiff board. Shift the origin to tha board's  
# centre  
for (ilog in 1:nrow(L)) {  
 ii = c(1:nrow(BJ))[BJ$ScionLogNumber == L[ilog, "ScionLogNumber"]]  
 pith.board = ii[order(BJ$moe[ii])][1]  
 BJ[ii, "x.minmoe"] = BJ[ii, "x"] - BJ[pith.board, "x"]  
 BJ[ii, "y.minmoe"] = BJ[ii, "y"] - BJ[pith.board, "y"]  
}  
BJ$R.minmoe = sqrt(BJ$x.minmoe^2 + BJ$y.minmoe^2)  
xyplot(R.minmoe ~ R, BJ)



plot of chunk unnamed-chunk-5

Is R.minmoe any better a predictor of twist?

par(mfcol = c(1, 2))  
plot(twist.rate ~ R, BJ)  
plot(twist.rate ~ R.minmoe, BJ)



plot of chunk unnamed-chunk-6

summary(lm(twist.rate ~ I(1/jitter(R)) + I(1/jitter(R.minmoe)), BJ))

##   
## Call:  
## lm(formula = twist.rate ~ I(1/jitter(R)) + I(1/jitter(R.minmoe)),   
## data = BJ)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.563 -0.326 0.035 0.353 2.747   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.46e-01 1.37e-02 -10.66 <2e-16 \*\*\*  
## I(1/jitter(R)) -1.68e-02 9.15e-03 -1.83 0.067 .   
## I(1/jitter(R.minmoe)) -9.59e-07 2.04e-05 -0.05 0.963   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.603 on 1936 degrees of freedom  
## (10 observations deleted due to missingness)  
## Multiple R-squared: 0.00173, Adjusted R-squared: 0.000697   
## F-statistic: 1.68 on 2 and 1936 DF, p-value: 0.187

Possibly R.minmoe is better is a better estimator of average distance to the pith for each board, but its not clear cut.

## 5. Add comment re inner v outer v total give similar models

lm2col = function(m, colname) {  
 s = summary(m)  
 a = s$coefficients[, 1]  
 a["r.squared"] = s$r.squared  
 a["RMSE"] = sqrt(mean(s$residuals^2, na.rm = TRUE))  
 df = data.frame(a)  
 names(df) <- c(colname)  
 return(df)  
}  
T = lm2col(lm(moe.bavg ~ swv.calibre + SED + heartVolFrac + ring10VolFrac, L),   
 "moe.all")  
T = cbind(T, lm2col(lm(moe.Inner.bavg ~ swv.calibre + SED + heartVolFrac + ring10VolFrac,   
 L), "moe.inner"))  
T = cbind(T, lm2col(lm(moe.Outer.bavg ~ swv.calibre + SED + heartVolFrac + ring10VolFrac,   
 L), "moe.outer"))  
T = cbind(T, lm2col(lm(abc.bavg ~ swv.calibre + SED + heartVolFrac + ring10VolFrac,   
 L), "abc.all"))  
T = cbind(T, lm2col(lm(abc.Inner.bavg ~ swv.calibre + SED + heartVolFrac + ring10VolFrac,   
 L), "abc.inner"))  
T = cbind(T, lm2col(lm(abc.Outer.bavg ~ swv.calibre + SED + heartVolFrac + ring10VolFrac,   
 L), "abc.outer"))  
library(pander)

## Attaching package: 'pander'  
##   
## The following object is masked from 'package:knitr':  
##   
## pandoc

pander(T, split.tables = Inf)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | moe.all | moe.inner | moe.outer | abc.all | abc.inner | abc.outer |
| **(Intercept)** | -9.846 | -6.692 | -9.84 | 25.89 | 23.47 | 28.07 |
| **swv.calibre** | 0.006449 | 0.005855 | 0.007425 | -0.005417 | -0.004603 | -0.006599 |
| **SED** | 0.003721 | -0.0006842 | -0.003291 | -0.0122 | -0.01187 | -0.008432 |
| **heartVolFrac** | -7.909 | -7.435 | -10.93 | 6.24 | 3.559 | 8.084 |
| **ring10VolFrac** | -3.737 | -4.708 | 3.036 | -1.137 | 0.2779 | -3.774 |
| **r.squared** | 0.8001 | 0.7429 | 0.6267 | 0.3604 | 0.3004 | 0.2009 |
| **RMSE** | 0.5837 | 0.6659 | 1.109 | 1.287 | 1.309 | 2.225 |

Models for the average board stiffness or ABC for all boards, the inner boards only and the outer boards only, while they differ considerably in their coeffcicients, are similar in their predictive abilities (RMSE, ) as evident in the table above.

## 6. Add conclusion re current set of metrics being cheap, easy, familiar and lousy but not close to potential if (i) better raw data, (ii) better metric extraction, (iii) better metric conception

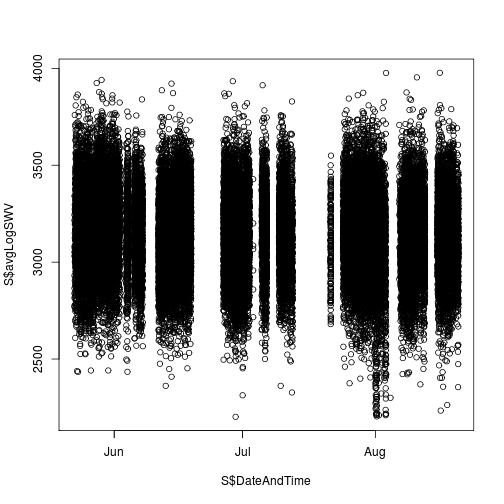
In the current study, the set of shape metrics extracted from the raw scan data have been selected based on previous experience and familiarity. They are cheap to compute and relatively easy to interpret, but should not be considered to be the 'best' possible shape-based predictors. There are numerous opportuinites for collecting better raw shape data (higher resolution, more precision), devising different descriptors and computing them more stably.

## 7. Rerun auto-correlation for stem velocity (based on avgLogSWV)

library(RODBC)  
ch = odbcConnect("KPPSWI", uid = "sa", pwd = "password12")  
S = sqlQuery(ch, "select \* from stems where recFrac>0.85 and avgLogSWV is not null and DateAndTime>'2014-05-01' and Forest not like '%9999%' order by DateAndTime")

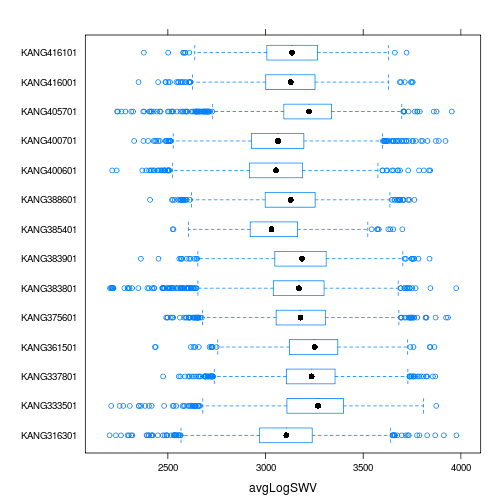
## Warning: closing unused RODBC handle 3

plot(S$DateAndTime, S$avgLogSWV)



plot of chunk unnamed-chunk-8

forest.count = table(S$Forest)  
many.log.forests = names(forest.count[forest.count > 1000])  
bwplot(Forest ~ avgLogSWV, S, subset = Forest %in% many.log.forests)

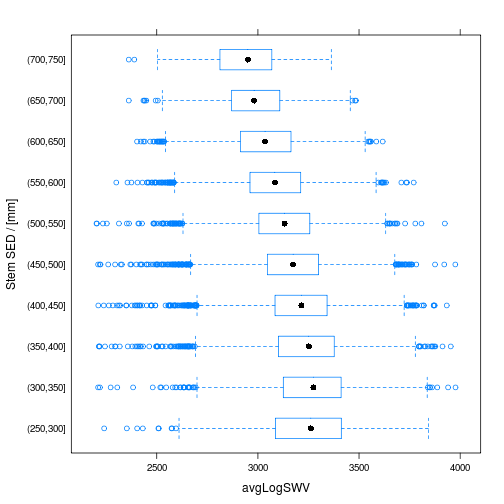


plot of chunk unnamed-chunk-8

sed.class = cut(S$stemLED, seq(250, 750, by = 50))  
table(sed.class)

## sed.class  
## (250,300] (300,350] (350,400] (400,450] (450,500] (500,550] (550,600]   
## 974 2780 5658 9295 10511 10142 7619   
## (600,650] (650,700] (700,750]   
## 4090 1529 409

bwplot(sed.class ~ avgLogSWV, S, ylab = "Stem SED / [mm]")



plot of chunk unnamed-chunk-8

Acf(S$avgLogSWV, type = "cor", demean = TRUE, lag.max = 250)

## Error: could not find function "Acf"

Pacf(S$avgLogSWV, lag.max = 250)

## Error: could not find function "Pacf"

# start with a linear model  
summary(lm(avgLogSWV ~ stemLED, S))

##   
## Call:  
## lm(formula = avgLogSWV ~ stemLED, data = S)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1086.8 -125.0 6.2 133.8 808.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.52e+03 4.50e+00 783.0 <2e-16 \*\*\*  
## stemLED -7.60e-01 9.07e-03 -83.8 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 195 on 53278 degrees of freedom  
## Multiple R-squared: 0.116, Adjusted R-squared: 0.116   
## F-statistic: 7.02e+03 on 1 and 53278 DF, p-value: <2e-16

# lets look at this another way...  
library(lme4)

## Loading required package: Matrix Loading required package: Rcpp

(lmer(avgLogSWV ~ (1 | Forest:sed.class), S))

## Linear mixed model fit by REML ['lmerMod']  
## Formula: avgLogSWV ~ (1 | Forest:sed.class)  
## Data: S  
## REML criterion at convergence: 701912  
## Random effects:  
## Groups Name Std.Dev.  
## Forest:sed.class (Intercept) 140   
## Residual 180   
## Number of obs: 53007, groups: Forest:sed.class, 207  
## Fixed Effects:  
## (Intercept)   
## 3128

### time series analysis stuff...  
library(forecast)

## Loading required package: zoo  
##   
## Attaching package: 'zoo'  
##   
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric  
##   
## Loading required package: timeDate This is forecast 5.5

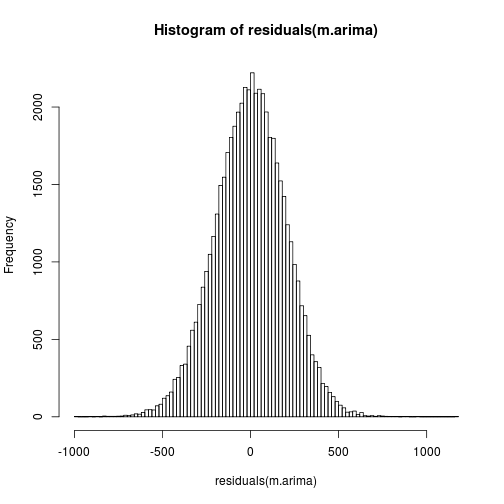
# https://www.otexts.org/fpp  
# http://www.statmethods.net/advstats/timeseries.html  
# http://www.stat.pitt.edu/stoffer/tsa3/tsa3EZ.pdf  
# http://stat565.cwick.co.nz/lectures/07-estimation.pdf  
# http://a-little-book-of-r-for-time-series.readthedocs.org/en/latest/src/timeseries.html  
# http://www.stat.pitt.edu/stoffer/tsa3/R\_toot.htm  
(m.arima = auto.arima(S$avgLogSWV, max.p = 200, max.q = 50, seasonal = FALSE,   
 stepwise = FALSE))

## Series: S$avgLogSWV   
## ARIMA(2,1,3) with drift   
##   
## Coefficients:

## Warning: NaNs produced

## ar1 ar2 ma1 ma2 ma3 drift  
## 0.359 0.455 -1.263 -0.147 0.414 -0.002  
## s.e. NaN NaN NaN NaN NaN 0.016  
##   
## sigma^2 estimated as 40102: log likelihood=-357951  
## AIC=715916 AICc=715916 BIC=715978

hist(residuals(m.arima), 100)

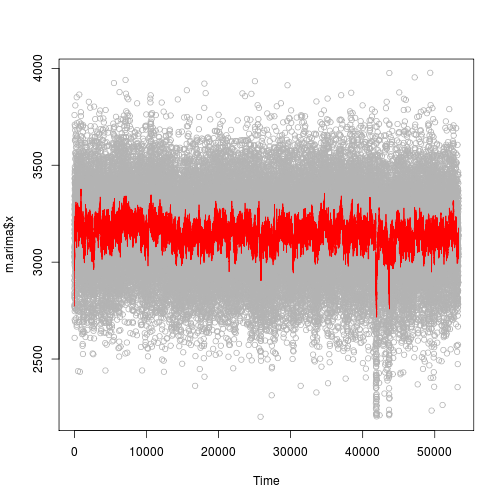


plot of chunk unnamed-chunk-8

summary(arima(S$avgLogSWV, xreg = S$Forest, order = c(2, 1, 3)))

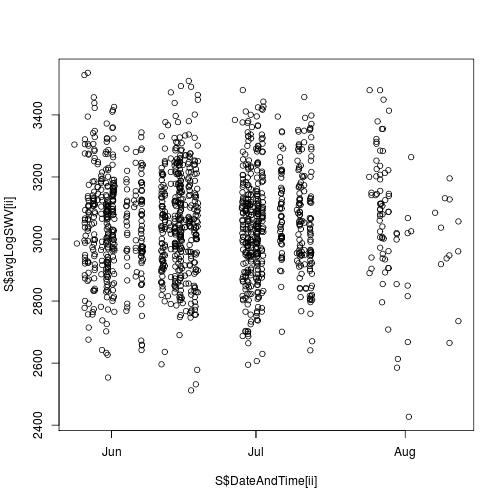
## Error: error in evaluating the argument 'object' in selecting a method for  
## function 'summary': Error in lm.fit(x, y, offset = offset, singular.ok =  
## singular.ok, ...) : 0 (non-NA) cases Calls: arima -> lm -> lm.fit

plot(m.arima$x, type = "p", cex = 0.3, col = "grey70")  
lines(m.arima$fitted, col = "red")



plot of chunk unnamed-chunk-8

# what if we confine our modelling to similar sized stems from the same  
# forest? KANG400701, (500,550] has the most data  
ii = !is.na(S$Forest) & !is.na(sed.class) & S$Forest == " KANG400701 " & sed.class ==   
 "(500,550]"  
plot(S$DateAndTime[ii], S$avgLogSWV[ii])

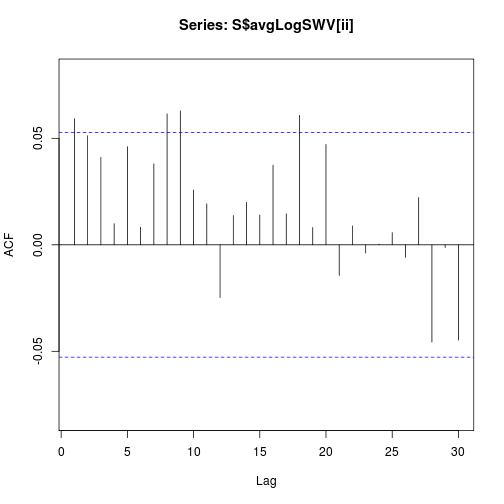


plot of chunk unnamed-chunk-8

sum(ii)

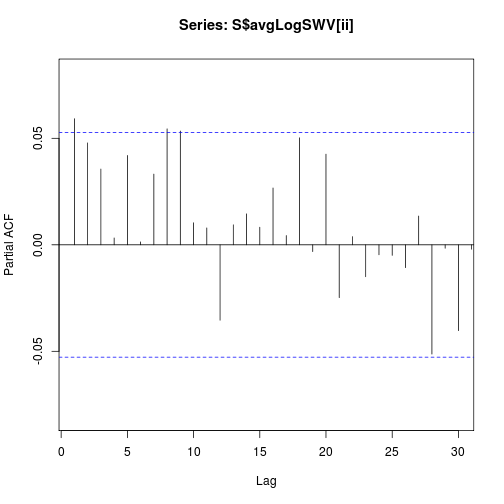
## [1] 1380

Acf(S$avgLogSWV[ii])



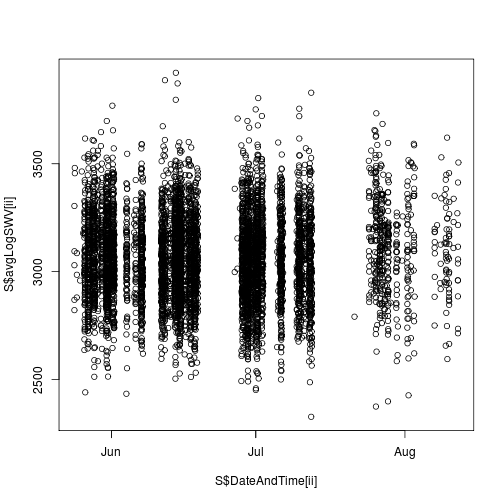
plot of chunk unnamed-chunk-8

Pacf(S$avgLogSWV[ii])



plot of chunk unnamed-chunk-8

# nothing in there at all  
  
ii = !is.na(S$Forest) & S$Forest == " KANG400701 "  
plot(S$DateAndTime[ii], S$avgLogSWV[ii])

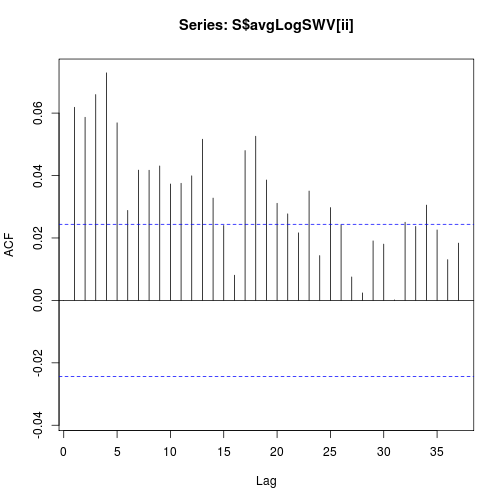


plot of chunk unnamed-chunk-8

sum(ii)

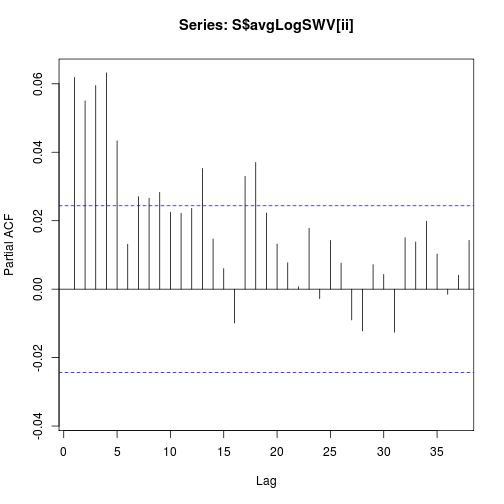
## [1] 6470

Acf(S$avgLogSWV[ii])



plot of chunk unnamed-chunk-8

Pacf(S$avgLogSWV[ii])



plot of chunk unnamed-chunk-8

sd(S$avgLogSWV)

## [1] 207.7

Means differ significantly between forests, but variance similar.

Stem SWV is influenced by diameter (LED), but again the effect is not strong.

Best ARIMA model reduces std error by only 7 m/s! (207 vs 200)

All of this suggests that the majority of variation in stem SWV is between individual stems.

Average log SWV can be used as a substitute for missing stem SWV, but the SWV of previously processed stems are poor predictors of the SWV of the current stem.

## 8. Table of TP vs FP for log stiffness and abc

How well can a linear model for average stiffness (moe) and warp (abc) serve to find the best or worst 10% of logs?

m.moe = lm(moe.bavg ~ swv.calibre + SED + heartVolFrac + ring10VolFrac, L)  
m.abc = lm(abc.bavg ~ swv.calibre + SED + heartVolFrac + ring10VolFrac, L)  
roc = function(meas, pred) {  
 TP = sum(pred & meas)  
 TN = sum(!pred & !meas)  
 FP = sum(pred & !meas)  
 FN = sum(!pred & meas)  
 return(list(TP = TP, TN = TN, FP = FP, FN = FN, TPR = TP/(TP + FN), FPR = FP/(TN +   
 FP)))  
}  
  
  
m = L$moe.bavg # measured  
p = predict(m.moe) # predicted  
crit = quantile(p, 0.9)  
R = roc(m > crit, p > crit)  
T = data.frame(crit = crit, TPR = R[["TPR"]], FPR = R[["FPR"]], row.names = c("moe.best"))  
  
crit = quantile(p, 0.1)  
R = roc(m < crit, p < crit)  
T = rbind(T, data.frame(crit = crit, TPR = R[["TPR"]], FPR = R[["FPR"]], row.names = c("moe.worst")))  
  
m = L$abc.bavg # measured  
p = predict(m.abc) # predicted  
crit = quantile(p, 0.1)  
R = roc(m < crit, p < crit)  
T = rbind(T, data.frame(crit = crit, TPR = R[["TPR"]], FPR = R[["FPR"]], row.names = c("abc.best")))  
  
crit = quantile(p, 0.9)  
R = roc(m < crit, p < crit)  
T = rbind(T, data.frame(crit = crit, TPR = R[["TPR"]], FPR = R[["FPR"]], row.names = c("abc.worst")))  
  
pander(T, split.table = Inf, digits = 2)

|  |  |  |  |
| --- | --- | --- | --- |
|  | crit | TPR | FPR |
| **moe.best** | 11 | 0.59 | 0 |
| **moe.worst** | 7.9 | 0.67 | 0.044 |
| **abc.best** | 3.5 | 0.32 | 0.027 |
| **abc.worst** | 6 | 0.95 | 0.67 |

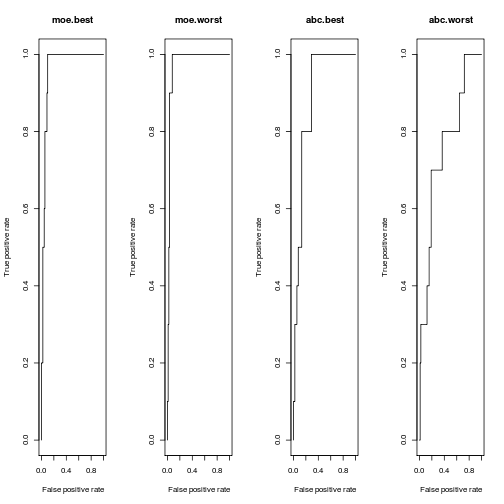
TPR: True Positive Rate. Ratio of logs that are predicted to meet the criteria compared to those that actually meet it. FPR: False Positive Rate. Ratio of logs that are predicted to not meet the criteria compared to those that actually do not meet it. Ideally TPR=1 and FPR=0. If FPR=TPR then the classifier is no better than random guessing.

The next plot presennts the so-called receiver operator curves for classification of best and worst logs using a linear model. These plots demonstrate the feasibility of finding elite logs (high stiffness and low warp), or of finding logs that will produce the least stiff lumber, but not of finding the worst logs for warp.

library(ROCR)

## Loading required package: gplots KernSmooth 2.23 loaded Copyright M. P.  
## Wand 1997-2009  
##   
## Attaching package: 'gplots'  
##   
## The following object is masked from 'package:stats':  
##   
## lowess

par(mfcol = c(1, 4))  
plot(performance(prediction(predict(m.moe), L$moe.bavg > quantile(L$moe.bavg,   
 0.9)), "tpr", "fpr"), main = "moe.best")  
plot(performance(prediction(-predict(m.moe), L$moe.bavg < quantile(L$moe.bavg,   
 0.1)), "tpr", "fpr"), main = "moe.worst")  
plot(performance(prediction(-predict(m.abc), L$abc.bavg < quantile(L$abc.bavg,   
 0.1)), "tpr", "fpr"), main = "abc.best")  
plot(performance(prediction(predict(m.abc), L$abc.bavg > quantile(L$abc.bavg,   
 0.9)), "tpr", "fpr"), main = "abc.worst")



plot of chunk unnamed-chunk-10

## 9. Scale sample logs to population using input metrics then tabulate: worst/best 5/10% of logs in sample = X% in pop. Only consider stiffness.

For each log in the sample set, find the number of logs in the population (or rather in the >190k butt logs seen to date) that have a velocity closer to this log than any other sample log:

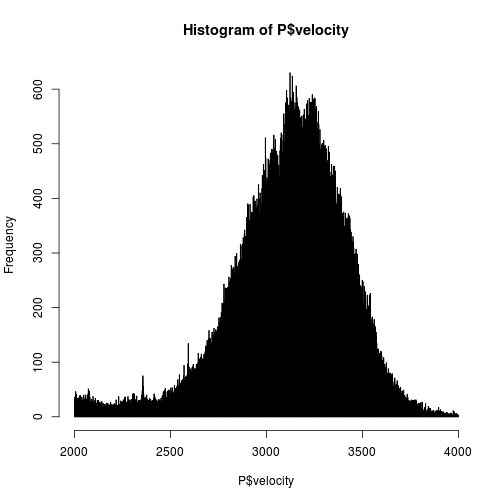
library(RODBC)  
ch = odbcConnect("KPPSWI", uid = "sa", pwd = "password12")  
P = sqlQuery(ch, "select DateAndTime, velocity from Phase2 where velocity>2000 and velocity<4000 and StartPos<1000")

## Warning: closing unused RODBC handle 4

nrow(P)

## [1] 187628

hist(P$velocity, 1000)

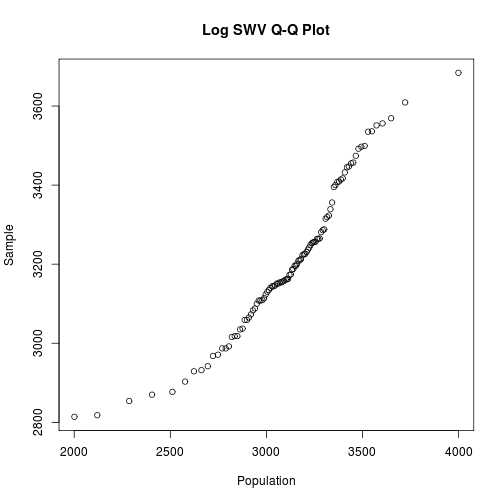


plot of chunk unnamed-chunk-11

swv = sort(L$swv.calibre)  
P$class = cut(P$velocity, breaks = c(-Inf, (swv[1:length(swv) - 1] + swv[2:length(swv)])/2,   
 Inf), labels = L$ScionLogNumber[order(L$swv.calibre)])  
table(P$class)

##   
## 44 93 73 70 31 54 6 12 2 62 51 23   
## 26038 2603 3548 1611 2710 4620 2515 1368 3479 2682 1999 1721   
## 52 82 22 46 81 75 74 26 38 67 50 5   
## 458 3368 2919 228 1830 2402 2886 2547 673 1579 2227 1715   
## 68 28 17 18 11 91 65 66 86 34 61 83   
## 2300 2714 1161 0 858 1912 2322 1713 1504 1422 562 274   
## 29 43 88 16 92 60 10 90 85 72 37 4   
## 812 868 268 566 297 605 572 547 312 1677 1385 1879   
## 33 78 1 69 89 19 36 27 57 96 63 24   
## 1872 1312 1088 563 1849 1099 546 1934 1661 255 823 1721   
## 58 42 3 35 100 20 25 53 95 98 79 15   
## 1443 1690 1359 1167 265 299 1048 1107 0 2405 2499 976   
## 64 56 14 49 45 59 80 87 77 7 99 84   
## 3425 3833 897 2301 3480 5855 4311 1264 690 518 733 1552   
## 47 94 32 97 76 30 8 55 21 48 39 13   
## 2475 1294 769 807 1331 2434 1310 443 2138 1784 778 854   
## 71 41 40 9   
## 752 1842 2639 3882

out = qqplot(P$velocity, L$swv.calibre, ylab = "Sample", xlab = "Population",   
 main = "Log SWV Q-Q Plot")

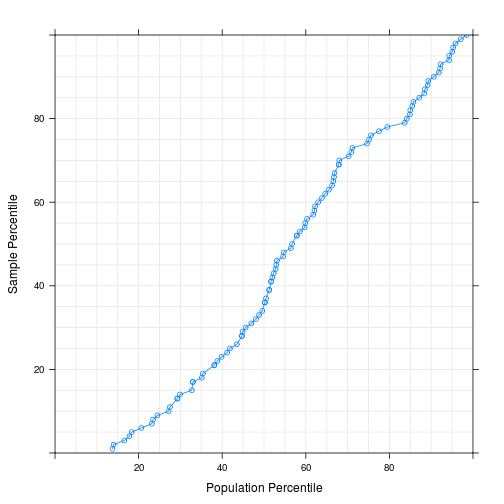


plot of chunk unnamed-chunk-11

F.s = ecdf(L$swv.calibre)  
F.p = ecdf(P$velocity)  
library(signal)

## Attaching package: 'signal'  
##   
## The following objects are masked from 'package:stats':  
##   
## filter, poly

xyplot(F.s(out[["y"]]) \* 100 ~ interp1(out[["x"]], F.p(out[["x"]]), out[["y"]]) \*   
 100, aspect = "iso", panel = function(...) {  
 panel.grid(h = 19, v = 19)  
 panel.xyplot(...)  
}, xlim = c(0, 100), ylim = c(0, 100), type = "b", xlab = "Population Percentile",   
 ylab = "Sample Percentile")



plot of chunk unnamed-chunk-11

# same as above but based on a normal model for the population  
# libaray(mixtools) fit=normalmixEM(P$velocity, k=3) # N(3153, 260.4)  
# fit=normalmixEM(P$velocity, k=6) # N(3150, 256.5)  
# fit=normalmixEM(P$velocity, k=2) # N(3165, 236.0) plot(fit, density=T)  
# xyplot(F.s(out[['y']]) ~ interp1(out[['x']],pnorm(out[['x']],mean=3150,  
# sd=256),out[['y']]), aspect='iso', panel=function(...) {  
# panel.grid(h=19,v=19) panel.xyplot(...)},  
# xlim=c(0,1),ylim=c(0,1),type='b') gives very similar results

The lowest log SWV in the sample is 2814 m/s, which is greater than the SWV of 10% of the logs in the population.

Based on an empirical distribution function for the log population:

* the worst 5% of the sample represent 15% of the population
* the worst 10% represent 25% of the population
* the best 10% represent 10% of the population
* the best 5% of the sample represent 7% of the population

At Hyne we want to avoid the skewed representation at the low stiffness end by not rushing the selection.