Supplement for: Pleistocene-aged stone artefacts from Jerimalai, East Timor: Long term conservatism in early modern human technology in island Southeast Asia

2015-11-10

# Introduction

This document is part of a research compendium that accompaies the analysis of stone artefacts from Jerimalai rockshelter reported in Marwick et al. "Pleistocene-aged stone artefacts from Jerimalai, East Timor: Long term conservatism in early modern human technology in island Southeast Asia".

This document has two parts. The first part explains the motivation for the Bayesian approach used in this study. The second part is organised with the same structure as the manuscript and will produce all of the quantitative results, data visualisations and tables found in the manuscript.

## Motivations for the Bayesian approach used in the paper

Statistical tests for this study were conducted using the Bayesian framework described in Krushcke (2011). Bayesian statistics are familiar to archaeologists in the context of radiocarbon dating because they provide formal methods for improving the calibration of radiocarbon ages by combining them with other data, such as stratigraphic information, that gives additional constraints to the chronological order of the dated samples (Ramsey, 2009). Beyond calibrating radiocarbon dates, there have been very few applications of Bayesian methods of statistical inference in archaeology (Buck and Litton, 1990a; Buck et al., 1996; Buck and Litton, 1990b; Dellaportas, 1998; Gowland and Chamberlain, 2002; Halekoh and Vach, 2004). Due to the rarity of Bayesian inference in archaeology more broadly, we present here an extended discussion to motivate its use in place of the classical Null Hypothesis Significance Tests (NHST). These classical tests, such as the t-test, chi-square and ANOVA, are based on frequentist inference, and are very common in the archaeological literature.

Although frequentist inference dominates the archaeological literature, we were motivated to explore alternatives by several concerns. Serious disadvantages of the frequentist approach have been identified in other disciplines over recent decades (some of the circa 500 publications on this topic include: Cohen, 1994; Gill, 1999; Guttman, 1985; Johnson, 1999; Lambdin, 2012; Nickerson, 2000; Schmidt, 1996; Stephens et al., 2007; Wagenmakers, 2007). The catalogue of problems and misunderstandings surrounding the use of NHST is extensive, so we here we only briefly review some of the most frequently noted objections before describing how Bayesian methods provide options for avoiding these problems in our specific case.

Many of the criticisms focus on misuse of the p-values that are generated by NHST methods (Halsey et al., 2015). For example, lower p-values are sometimes interpreted as indicating greater significance (an effect size statistic is necessary to obtain this information; Gliner et al., 2002). Further misunderstandings include that the p-value gives the odds that a research hypothesis is correct, that a result will replicate, that the null hypothesis is true, and that statistical significance indicated by the p-value is equivalent to scientific significance (Carver, 1993; Cohen, 1994; Lambdin, 2012). Even when the NHST results are interpreted correctly, others have pointed out that NHST are often under-powered, and that small differences in estimates of population parameters from large samples, no matter how scientifically insignificant, will yield significant NHST results (Johnson, 1999; Nix and Barnette, 1998). While some of these problems can be mitigated by including related statistical measures (such as effect sizes and confidence intervals), there have been claims that researchers should abandon NHST as a method of statistical inference (Loftus, 1996) or be forbidden from using them in publication (Fidler et al., 2004; Shrout, 1997). Following these calls, the journal Basic and Applied Social Psychology announced a ban on p-values in February 2015, arguing they were frequently used to support lower-quality research (Trafimow and Marks, 2015).

Although several alternatives exist to frequentist inference, our choice of a Bayeasian approach is motivated by its suitability to the specific details of this analysis. We were especially motivated by the relative conceptual and computational simplicity of conducting Bayesian analyses, compared to the alternatives (ie. likelihood-based statistics, and Akaikean-Information Criterion-based statistics). It is noteworthy that among philosophers of science, both supporters and critics consider Bayesianism to be dominant view in their field, and the paradigm has been central to recent developments in disciplines as philosophy, statistics, ecology, and computer science (Bandyopadhyay and Forster, 2009).

There are two specific details of this analysis that are relevant to our choice of a Bayesian approach. First is that Bayesian methods generally provide a more coherent basis for working with data from non-repeatable events such as an archaeological excavation (compared to, for example agricultural field trials, cf. Fisher 1921). Bayesian methods use the data at hand to produce posterior probability statements about distributions of parameters and hypotheses (Puga et al., 2015). NHST procedures do the reverse, computing the probability of an event as a point value indicating its relative frequency of occurrence in an infinite sequence of repeated experiments. In this way, NHST methods use a null hypothesis to assess the plausibility of the observed data (and more extreme data sets that were not observed but might have been with additional sampling or experiments), with another step of reasoning required to either reject or fail to reject the null hypothesis (Jackman, 2009). This means that Bayesian inference is based on the actual occurring data, not all possible data sets that might have occurred in an infinite number of hypothetical repetitions of the study (Bolstad, 2007). We consider Bayesian inference to be a conceptually simpler and more direct approach where we compute the probability of our hypotheses, not our data, as the NHST framework does.

Although Bayesian methods have intuitive appeal, they have been criticised as subjective and arbitrary because typically the analyst must choose a prior distribution over the unknown parameters (eg. mean, standard deviation, etc.) of the model to capture their beliefs about the situation before working with the data. Once the data are collected, Bayes rule is used to combine the prior distribution with the data to compute a posterior distribution of the unknown parameters. Most current applications formalise the choice of prior distributions by either following expert recommendations, estimating the distribution from the observed data (known as an empirical Bayesian approach, Carlin and Louis, 2011), using pre-existing data sets to generate priors (Anholt et al., 2000; Carlin and Louis, 2011; McCarthy and Masters, 2005), or using uninformative or weakly informative prior distributions (such as the uniform distribution) that do not give strong prior probability to any hypotheses about the data (Burnham and Anderson, 2002; Efron, 2013). We describe our choice of prior and likelihood functions in the methods section below. We also have made freely available all the raw data and R programming code used to compute the analyses presented here, to facilitate close inspection and reuse of our quantitative methods.

## Bayesian one-way ANOVA and Bayesian Poisson exponential ANOVA

We used two types of Bayesian methods, a one-way ANOVA for comparing groups of metric data (such as artefact measurements over different phases of occupation, where an ANOVA is the common NHST), and a Poisson exponential ANOVA for contingency table analysis (such as artefact counts, where a chi-square is the common NHST). Our supplementary information includes further discussion of the motivation for using a Bayesian approach, as well as an R package that makes these methods available for use with other datasets. For an extended technical description and graphical models of these methods, see Krushcke (2011). Our Bayesian one-way ANOVA takes the metric predicted variable (eg. artefact length or mass) and considers how it is deflected by each of the nominally scaled predictor variables (eg. depositional phase). These deflection parameters are the primary interest, and following Gelman (2005, 2006) we apply a hierarchical (or multilevel) model with a folded-t prior distribution (because it does not have infinite density near zero, so it behaves well when group-level variance is at or near zero) on the standard deviation, a uniform (or flat) distribution on the variance between levels (as an uninformative prior) and a fat-tailed normal distribution on the likelihood (to better accommodate outliers in the data). The parameters of these distributions are derived from the observed data. The combination of the specific types of prior distributions used here, and that their parameters come from the observed data, mean that this Bayesian ANOVA uses weakly informative priors that are not intended to represent our actual prior state of knowledge (which is small in this case) but rather to constrain the posterior distribution, to an extent allowed by the data (Gelman 2006).

For the contingency table analysis, a Poisson likelihood distribution is used as an exponential link function from an underlying ANOVA model. The term ANOVA does not imply groups of a nominal measurement variable in our observed data (which are counts in this case). Instead it refers to the general ANOVA-like approach of comparing the distributions of cell frequencies that we produce when predicting the cell frequencies. Where the chi-square computes a single estimated value per cell of the table, the Bayesian approach generates a distribution of values for each cell of the table. Comparing these distributions is analogous to two-way ANOVA, and investigating the relationships within and between variables in the table is analogous to main effects and interaction contrasts in ANOVA. The Poisson distribution is well-suited as the likelihood function for count data because it returns only non-negative integer values and is widely used to model discrete occurrences in time or across space (Sadiku and Tofighi 1999, Jackman 2009). This method also uses the folded-t prior, like the one-way ANOVA above, and thus has a weakly informative prior that reflects our low degree of prior knowledge about these assemblages.

To obtain a result from these tests we computed the posterior distributions, or evaluated the outcome of combining the data with our beliefs about the processes that produced the data (weakly held, in this case, so that the data contribute much more information than the prior about the parameters of the posterior). The general process of obtaining the posterior distributions is to combine, or integrate, the prior and likelihood distributions. We used Markov chain Monte Carlo (MCMC) methods to approximate the integral by taking a very large number of independent samples from the distributions to approximate the distribution of the product of the prior and the likelihood. From this approximated distribution we computed descriptive statistics of the posterior distribution to interpret the substantive significance of the output.

Our results from these Bayesian tests are reported here as highest probability density intervals (HDIs), with more extensive output presented in the supplementary materials. The HDI is the range of values that contain 95% of the values in the posterior distribution produced by the MCMC sampling. Our supplementary materials also include the equivalent null-hypothesis-significance-test for direct comparison by readers who are unfamiliar with Bayesian tests. The supplementary materials also include the raw data used for all tests so that others may reuse the measurements in their own tests and combine with other datasets.

### References

Anholt, B.R., Werner, E., Skelly, D.K., 2000. Effect of food and predators on the activity of four larval ranid frogs. Ecology 81, 3509-3521.

Bandyopadhyay, P., Forster, M., 2009. Introduction to the Philosophy of Statistics, Handbook for the Philosophy of Science: Philosophy of Statistics. Elsevier, Amsterdam, pp. 1-50.

Bolstad, W.M., 2007. Introduction to Bayesian statistics. John Wiley & Sons, New York.

Buck, C., Litton, C., 1990a. A computational Bayes approach to some common archaeological problems, Proceedings of the 2002 Computer Applications in Archaeology Conference, <http://proceedings.caaconference.org/files/1990/15_Buck_Litton_CAA_1990.pdf>.

Buck, C.E., Cavanagh, W.G., Litton, C.D., 1996. Bayesian approach to interpreting archaeological data. Wiley Chichester England.

Buck, C.E., Litton, C., 1990b. A computational Bayes approach to some common archaeological problems. Computer applications and quantitative methods in archaeology, 93-99.

Burnham, K.P., Anderson, D.R., 2002. Model selection and multimodel inference: a practical information-theoretic approach. Springer-Verlag, New York.

Carlin, B.P., Louis, T.A., 2011. Bayesian methods for data analysis. CRC Press.

Carver, R.P., 1993. The case against statistical significance testing, revisited. Journal of Experimental Education 61, 287-292.

Cohen, J., 1994. The earth is round (p<. 05). American Psychologist 49, 997-1003.

Dellaportas, P., 1998. Bayesian classification of neolithic tools. Applied Statistics 47, 279-297.

Efron, B., 2013. Bayes theorem in the 21st century. Science 340, 1177-1178.

Fidler, F., Geoff, C., Mark, B., Neil, T., 2004. Statistical reform in medicine, psychology and ecology. The Journal of Socio-Economics 33, 615-630.

Fisher, R.A., 1921. Studies in crop variation. I. An examination of the yield of dressed grain from Broadbalk. The Journal of Agricultural Science 11, 107-135.

Gelman, A. (2005). "Analysis of variancewhy it is more important than ever." The Annals of Statistics 33(1): 1-53.

Gelman, A. (2006). "Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper)." Bayesian analysis 1(3): 515-534.

Gill, J., 1999. The insignificance of null hypothesis significance testing. Political Research Quarterly 52, 647-674.

Gliner, J.A., Leech, N.L., Morgan, G.A., 2002. Problems with null hypothesis significance testing (NHST): what do the textbooks say? The Journal of Experimental Education 71, 83-92.

Gowland, R.L., Chamberlain, A.T., 2002. A Bayesian approach to ageing perinatal skeletal material from archaeological sites: implications for the evidence for infanticide in Roman-Britain. J Archaeol Sci 29, 677-685.

Guttman, L., 1985. The illogic of statistical inference for cumulative science. Applied stochastic models and data analysis 1, 3-9.

Halekoh, U., Vach, W., 2004. A Bayesian approach to seriation problems in archaeology. Computational Statistics & Data Analysis 45, 651-673.

Halsey, L.G., Curran-Everett, D., Vowler, S.L., Drummond, G.B., 2015. The fickle P value generates irreproducible results. Nature Methods 12, 179-185.

Jackman, S., 2009. Bayesian analysis for the social sciences. John Wiley & Sons, Melbourne.

Johnson, D.H., 1999. The insignificance of statistical significance testing. The Journal of Wildlife Management 63, 763-772.

Krushcke, J.K., 2011. Doing Bayesian Data Analysis: A Tutorial with R and BUGS. Academic Press, New York.

Lambdin, C., 2012. Significance tests as sorcery: Science is empiricalsignificance tests are not. Theory & Psychology 22, 67-90.

Loftus, G.R., 1996. Psychology will be a much better science when we change the way we analyze data. Current directions in psychological science, 161-171.

McCarthy, M.A., Masters, P.I.P., 2005. Profiting from prior information in Bayesian analyses of ecological data. Journal of Applied Ecology 42, 1012-1019.

Nickerson, R.S., 2000. Null hypothesis significance testing: A review of an old and continuing controversy. Psychological Methods 5, 241-301.

Nix, T.W., Barnette, J.J., 1998. The data analysis dilemma: Ban or abandon. A review of null hypothesis significance testing. Research in the Schools 5, 3-14.

Puga, J.L., Krzywinski, M., Altman, N., 2015. Points of significance: Bayesian statistics. Nature Methods 12, 377-378.

Ramsey, C.B., 2009. Bayesian analysis of radiocarbon dates. Radiocarbon; Vol 51, No 1 (2009).

Reimer, P.J., Bard, E., Bayliss, A., Beck, J.W., Blackwell, P.G., Ramsey, C.B., Buck, C.E., Cheng, H., Edwards, R.L., Friedrich, M., 2013. IntCal13 and Marine13 radiocarbon age calibration curves 050,000 years cal BP. Radiocarbon 55, 1869-1887.

Sadiku, M. N. O. and M. R. Tofighi (1999). "A tutorial on simulation of queueing models." International Journal of Electrical Engineering Education 36(2): 102-120.

Schmidt, F.L., 1996. Statistical significance testing and cumulative knowledge in psychology: Implications for training of researchers. Psychological Methods 1, 115.

Shrout, P.E., 1997. Should significance tests be banned? Introduction to a special section exploring the pros and cons. Psychological Science 8, 1-2.

Stephens, P.A., Buskirk, S.W., del Rio, C.M., 2007. Inference in ecology and evolution. Trends in Ecology & Evolution In Press, Corrected Proof.

Trafimow, D., Marks, M., 2015. Editorial. Basic and Applied Social Psychology 37, 1-2.

Wagenmakers, E.-J., 2007. A practical solution to the pervasive problems ofp values. Psychonomic bulletin & review 14, 779-804.

## Reproducing the results in the paper

The source file for this document is an R markdown document, which means that it is a plain text file that contains marked-up text (such as this paragraph), and chunks of the R programming language in between paragraphs of plain text. This file may be viewed in any text editor, and executed to run the code using R version 3.2.0.

This section of the document is organised with the same structure as the manuscript and will produce all of the quantitative results, data visualisations and tables found in the manuscript. This document is an R markdown document, which means that it is a plain text file that contains marked-up text (such as this paragraph), and chunks of the R programming language in between paragraphs of plain text. This file may be viewed in any text editor, and executed to run the code using R version 3.2.0. When the document is executed, a HTML file is produced that contains the tables, plots and other statistical output generated by the code (if you see plots below, you are reading this HTML file). All of the analysis code is present in both the R markdown and HTML files, but some lines of setup code have been hidden from the HTML file to improve readabiltiy.

The purpose of providing this supplement is to enable the reader to more thoroughly evaluate the reliability of our work through close inspection the quantitiative methods we used in the paper, and to enable computational reproduciblity of our methods to facailite their application to other research projects. Note that the code presented here has been developed and tested specifically for this analysis, and is not yet intended as a general purpose tool.

## Code and Dependencies

The exact version of the code that produced the data and figures in the published paper is archived at [<http://dx.doi.org/10.6084/m9.figshare.985406>](http://dx.doi.org/10.6084/m9.figshare.985406). The development version of this code can be found at [<https://github.com/benmarwick/Pleistocene-aged-stone-artefacts-from-Jerimalai>--East-Timor](https://github.com/benmarwick/Pleistocene-aged-stone-artefacts-from-Jerimalai--East-Timor). Note that the code in the development version on github may have changed since publication and may not produce exactly the same output as found in the published paper.

The code should run on a typical personal computer (Windows/OSX/Linux) that can run R and can install R packages from the internet. The specific software dependencies for the code included here are listed in the code chunk above this text and the DESCRIPTION file of the R package that this supplement file is contained in (the R package name is JerimalaiStoneArtefacts).

Managing the dependencies can be tedious, and we have no control over how they will change in the future. In an effort to capture the entire computational environment that this analysis was developed and conducted in, and protect against changes in the dependant software packages, we provide a Docker image as a lightweight virtual machine that is the actual environment in which we developed, tested and ran the code. The Docker image contains all the necessary software, code and data already installed, so no further configuration is required. We also include a Dockerfile as a record of the instructions used to build the Docker image, this can be found in the docker/ directory in the research compendium. To launch the Docker image for this project, first, [install Docker](https://docs.docker.com/installation/), then run Docker and at the Docker prompt, enter:

docker run -dp 8787:8787 benmarwick/jerimalaistoneartefacts

Then open a web broswer at the following URL to access RStudio (username and password are "rstudio"):

http://localhost:8787/ ## Linux users  
http://192.168.59.103:8787/ ## OSX, Windows users

Once logged in, the Files pane (bottom right) will show the manuscript/ directory where you can find this document and execute it. More information about using RStudio in Docker is avaiable at the [Rocker](https://github.com/rocker-org) [wiki](https://github.com/rocker-org/rocker/wiki/Using-the-RStudio-image) pages.

## Data

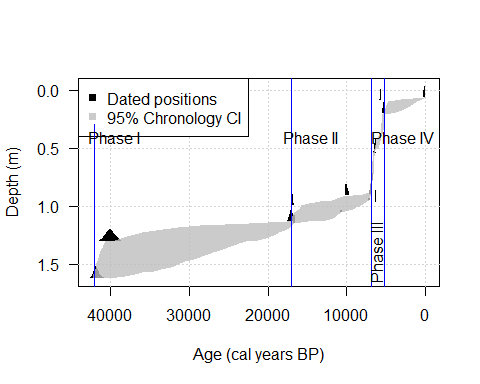
All of the data needed to reproduce the results presented in the paper are included in this compendium as CSV files (ie. plain text comma separated variables, readable in any text editor). They can be found in the data/ directory and are made available with a [CC-0 licence](https://creativecommons.org/publicdomain/zero/1.0/). The code chunk below loads all of the data files into the R working environment (assuming you have the working directory set to the location of this supplement file).

# dates  
dates <- read.csv("data/Jerimalai\_dates\_Square\_B.csv", as.is = TRUE)  
# complete flake data  
flakes <- read.csv("data/JB\_Chert\_Flakes\_and\_Retouch.csv")  
# core type data  
core\_types <- read.csv("data/Jerimalai\_cores\_techno\_metrics.csv")  
# spit depths  
depths <- read.csv("data/Jerimalai\_spit\_depths\_Square\_B.csv")  
# sediment volumes  
vols <- read.csv("data/Artefact densities with soil volumes Sq B.csv", skip = 1)  
# all artefacts  
all <- read.csv("data/Jerimalai\_All\_Artefacts\_Square\_B.csv")  
# techno types  
cores <- read.csv("data/Jerimalai\_tech\_table\_cores.csv")  
types <- na.omit(read.csv("data/Jerimalai\_tech\_table\_types.csv"))  
retouch <- read.csv("data/Jerimalai\_tech\_table\_retouch.csv")  
features <- read.csv("data/Jerimalai\_tech\_table\_features.csv")  
ground <- read.csv("data/Jerimalai\_tech\_table\_ground.csv")  
# retouch indices  
retouch\_indices <- read.csv("data/Jerimalai\_retouch\_indices.csv")

# Chronology of the excavated deposit

The code chunk below reproduces Figure 2, the depth-age distribution for radiocarbon dates from Jerimalai square B.

# ages <- BchronCalibrate(ages = dates$age,  
 # ageSds = dates$error,  
 # positions = dates$depth\_bs,   
 # calCurves = rep("intcal13",   
 # length(dates$age)))  
# show tables of calibrated age ranges for each date  
# summary(ages)  
  
# plot(ages, xlab='Age (cal years BP)', withPositions = TRUE)  
  
  
ages\_predict = Bchronology(ages = dates$age,  
 ageSds = dates$error,  
 positions = dates$depth\_bs,   
 ids = dates$lab\_code,  
 calCurves = rep("intcal13", length(dates$age)))  
  
# save plot  
png("figures/fig\_2-Jeremalai-dates.png",   
 width = 200,   
 height = 120,   
 units = "mm", res = 100)  
  
bchron\_plot <- function(){  
plot(ages\_predict,  
 main="",  
 xlab='Age (cal years BP)',  
 ylab='Depth (m)',  
 las=1 ) #,   
 #asp=0.6)  
  
# add phases  
  
phases <- data.frame(phase = 1:4,  
 start = c(42, 17, 6.9, 5.3),  
 end = c(35, 9, 5.5, 0 ))  
  
phase\_lines <- (phases$start \* 1000)  
line\_height <- c(1.7, 0.3, -0.1, 0.4)  
  
lines(rep(phase\_lines[1],2), line\_height[1:2], col = "blue")  
lines(rep(phase\_lines[2],2), line\_height[c(1,3)], col = "blue")  
lines(rep(phase\_lines[3],2), line\_height[c(1,3)], col = "blue")  
lines(rep(phase\_lines[4],2), line\_height[c(1,3)], col = "blue")  
  
text(phase\_lines[1] - 2500, line\_height[4], labels = "Phase I")  
text(phase\_lines[2] - 2500, line\_height[4], labels = "Phase II")  
text(phase\_lines[3] - 600, line\_height[4]+1, labels = "Phase III", srt = 90)  
text(phase\_lines[4] - 2500, line\_height[4] , labels = "Phase IV")  
}  
bchron\_plot()  
# end saving plot  
dev.off()  
  
# combine all the calibrated dates into a single plot (using results = 'hide' to hide the progress bar)  
# ages\_densities <- BchronDensity(ages = dates$age,  
# ageSds = dates$error,  
#   
# calCurves = rep("intcal13", length(dates$age)))  
  
# plot(ages\_densities, xlab='Age (cal years BP)', withPositions = TRUE)  
  
# and show the plot when this document is knited  
bchron\_plot()



# Results: Raw materials

The code chunk below computes the percentages of the two major cortex types for chert flakes, and displays the result in a table.

cortex\_type <- as.data.frame(table(flakes$Cortype))[-1,]  
cortex\_type$prop <- round(prop.table(cortex\_type$Freq),2)  
names(cortex\_type) <- c("Cortex\_type", "Frequency", "Proportion")  
kable(cortex\_type, caption = "Cortex type among chert flakes")

Cortex type among chert flakes

|  |  |  |  |
| --- | --- | --- | --- |
|  | Cortex\_type | Frequency | Proportion |
| 2 | Angul | 239 | 0.65 |
| 3 | Round | 131 | 0.35 |

The chunk above also generates the values that are found in this sentence in the manuscript: "The chert artefacts have a combination of rounded cortex (35%) and angular cortex (65%.")

The code chunk below subsets the stone artefact data so that we only include flakes without any missing mass data. We then assign each flake to a depositional phase based on the spit it was recovered from. There is no visual output from this chunk but the values are stored and used in the following chunks.

# omit rows with blanks or NAs  
flakes <- flakes[!(flakes$Weight == "" | is.na(flakes$Weight)), ]  
  
# put depths on lithic data  
flakes$depth <- depths$Depth.bs..m[match(flakes$Spit,depths$Spit.no)]  
  
# omit rows with blanks or NAs... again  
flakes <- flakes[!(flakes$depth == "" | is.na(flakes$depth)), ]  
  
flakes$phase <- ifelse(flakes$Spit > 3 & flakes$Spit <= 20, 4,  
 ifelse(flakes$Spit >= 21 & flakes$Spit <= 39, 3,  
 ifelse(flakes$Spit >= 40 & flakes$Spit <= 48, 2,  
 ifelse(flakes$Spit >= 49 & flakes$Spit <= 69, 1, NA))))  
# check if any NA  
check <- unique(flakes$phase)  
  
# again...  
phases <- data.frame(phase = 1:4,  
 start = c(42, 17, 6.9, 5.3),  
 end = c(35, 9, 5.5, 0 ))  
  
# here's a function to assign phases based on spit numbers  
 makephases <- function(x) {ifelse(x >= 3 & x <= 20, 4,  
 ifelse(x >= 21 & x <= 39, 3,  
 ifelse(x >= 40 & x <= 48, 2,  
 ifelse(x >= 49 & x <= 69, 1, NA))))}  
  
flakes <- flakes[!(is.na(flakes$Spit)),]  
flakes$phase <- makephases(flakes$Spit)  
# check if any NA, don't want any, should return all TRUE  
# is.na(unique(flakes$phase)) %in% FALSE  
# get phase durations  
  
phases$duration <- with(phases, start - end)

The code chunk below generates a table that summarises the frequencies of the main raw materials in each depositional phase. A raw material is considered dominant here if there are more than ten peices in a phase.

# raw material  
raw <- dcast(flakes, Material ~ phase)   
# remove row with no raw material  
raw <- raw[-10,]  
rownames(raw) <- raw[,1]  
# get rid of rows with no value  
raw <- raw[rownames(raw) != "",]  
# remove col of NA  
raw <- raw[,colnames(raw) != "NA"]  
raw <- raw[,-1]  
# subset dominant raw materials  
dom <- raw[rowSums(raw) > 10,]  
colnames(dom) <- c("phase 1", "phase 2", "phase 3", "phase 4")  
kable(dom, caption = "Frequencies of dominant raw materials by depositional phase")

Frequencies of dominant raw materials by depositional phase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | phase 1 | phase 2 | phase 3 | phase 4 |
| Chert | 432 | 333 | 1048 | 801 |
| Quartz | 0 | 1 | 8 | 6 |
| Quartzite | 2 | 2 | 14 | 10 |
| Silcrete | 7 | 8 | 11 | 13 |
| Unknown | 6 | 4 | 0 | 4 |
| Volcanic | 28 | 23 | 46 | 75 |

The code chunk below uses the Bayesian Poisson exponential ANOVA to compute the probabilities of any credible interactions between raw material frequencies and depositional phase. The goal is to determine if there are any significant changes in the use of raw materials for stone artefacts over time.

data <- melt(as.matrix(dom), varnames=c("raw\_material", "phase"), value.name="Freq")  
myDataFrame = data  
yName=names(myDataFrame)[3] # Freq  
x1Name=names(myDataFrame)[2] # phase   
x2Name=names(myDataFrame)[1] # raw material  
x1contrasts = list(   
 list( c("phase 1") , c("phase 2") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("phase 2") , c("phase 3") , compVal=0.0 , ROPE=c(-0.1,0.1)) ,  
 list( c("phase 3") , c("phase 4") , compVal=0.0 , ROPE=c(-0.1,0.1))   
 # list( c("phase 4") , c("phase 5") , compVal=0.0 , ROPE=c(-0.1,0.1))  
 )  
numSavedSteps = 12000 # MCMC parameters  
thinSteps = 10 # MCMC parameters  
  
mcmcCoda = genMCMC\_cont\_table( datFrm=myDataFrame ,   
 yName=yName , x1Name=x1Name , x2Name=x2Name ,  
 numSavedSteps=numSavedSteps , thinSteps=thinSteps ,   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 10:04:36 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 304  
  
WARNING: Unused variable(s) in data table:  
ySum  
  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 1000  
-------------------------------------------------| 1000  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 2000  
-------------------------------------------------| 2000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . . . . . . . Failed to set trace monitor for delta  
Node not found  
. Updating 40000  
-------------------------------------------------| 40000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . Updating 0  
. Deleting model  
.   
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
summaryInfo = smryMCMC\_cont\_table( mcmcCoda ,   
 datFrm=myDataFrame , x1Name=x1Name , x2Name=x2Name ,  
 x1contrasts=x1contrasts   
 )  
  
# # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_cont\_table( mcmcCoda ,   
# datFrm=myDataFrame , yName=yName , x1Name=x1Name ,   
# x2Name=x2Name ,  
# x1contrasts=x1contrasts )  
  
# subset summaryInfo to show HDI interval for interactions  
HDI\_intervals\_for\_interactions <- summaryInfo[c((nrow(summaryInfo)-length(x1contrasts)):nrow(summaryInfo)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions,3), caption = "Highest density intervals (95%) for the posterior distributions of the interactions between phases and raw material frequencies.")

Highest density intervals (95%) for the posterior distributions of the interactions between phases and raw material frequencies.

|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| a1a2SD | 0.098 | 0.732 |
| phase 1.v.phase 2 | -0.249 | 0.534 |
| phase 2.v.phase 3 | -1.226 | -0.473 |
| phase 3.v.phase 4 | -0.364 | 0.245 |

The code chunk above provides psoterior distributions to evaluate the credibility of differences in raw material frequences over time. The posterior distributions for the phase interactions are close to zero, indicating a small effect. Among the contrast of the phases, all HDIs exclude zero except for phase 4 v. phase 5. This indicates that there are credibly different frequencies of raw material between each phase except for four and five. This is likely are result of small changes in the low frequencies of quartz, quartzite and silcrete in the earlier phases, as shown in the phase by raw material table above.

The code chunk below computes a NHST equivalent to the above Bayesian test, in this case a chi-square test. We also compute Cramer's V, a measure of association for nominal variables that ranges from 0 (no association between the variables) to 1 (strong association between the variables).

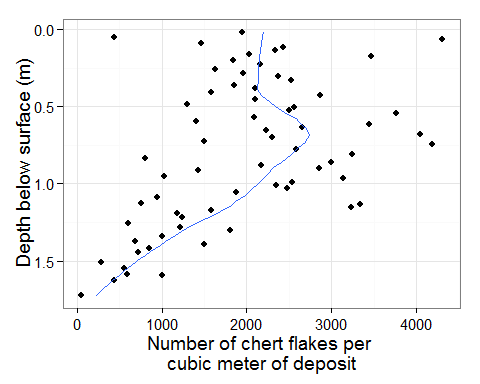
# here is the frequentist equivalent   
raw\_material\_by\_phase\_nhst <- chisq.test(dom)  
cramers\_V <- assocstats(as.matrix(dom))

The code chunk above returns a chi-squared value of 40.4388505 and a p-value of 3.89208710^{-4}, indicating a signficant difference in raw material frequencies by phase. However, the Cramer's V value of 0.0683899 indicates that the effect size is extremely small. We interpret this result to mean that there is no substantial significance in the differences in raw materials frequencies by phase.

# Results: Discard rates

The code chunk below generates the figure that shows discard rates of chert artefacts over time at Jerimalai square B. Each point is an excavation unit. The blue line is a locally weighted regression line (span = 0.4) to aid in visualising the trend of increased discard in the upper part of the deposit.

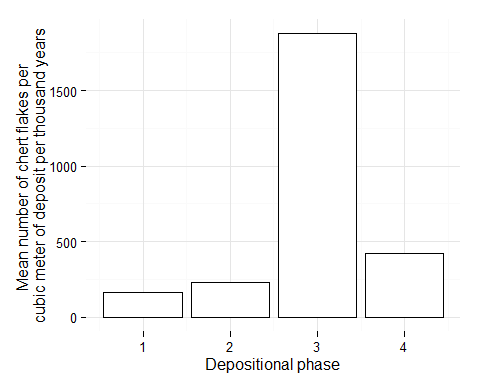
# discard rates  
discard <- aggregate(Weight ~ depth + Spit, flakes, length)  
# sediment volumes: put volumes on  
discard$sedvol <- vols$Soil[match(discard$Spit, vols$X)]  
# put spit thickesses on  
discard$thick <- c(0.018, diff(discard$depth)) # add first value from depth\_and\_dates.xsl  
# compute artefacts per kg of sediment  
discard$kgsed <- with(discard, Weight / sedvol) # weight is count of artefacts that have a weight  
# compute artefact per cubic meter (spit thickess)  
discard$cubmet <- with(discard, Weight / thick)  
# seems we have an unusually extreme value in spit 34  
# omit - perhaps a data collection typo  
discard <- discard[discard$Spit != 34, ]  
# Plot  
ggplot(discard, (aes(depth, cubmet))) +  
 geom\_point() +  
 stat\_smooth(span = 0.5, se = FALSE) +  
 xlab("Depth below surface (m)") +  
 ylab("Number of chert flakes per \ncubic meter of deposit") +  
 coord\_flip() +  
 scale\_x\_reverse()



# save plot  
jhe\_90mm\_ggsave("figures/fig\_3\_Jeremalai-flake-discard.png")

The code chunk below the generates the figure showing discard rates of chert artefacts per depositional phase at Jerimalai square B. This aggregates the individual excavation units.

# plot artefacts/cubic meter/1000 years by phase, get phase number for each spit  
# this is the most sensible option  
discard$phase <- makephases(discard$Spit)  
discard\_agg <- aggregate(cubmet ~ phase, discard, mean)  
discard\_agg$cubmetperkyr <- discard\_agg$cubmet / phases$duration  
ggplot(discard\_agg, (aes(phase, cubmetperkyr))) +  
 geom\_bar(stat="identity", colour = "black", fill = "white") +  
 theme\_minimal() +  
 xlab("Depositional phase") +  
 ylab("Mean number of chert flakes per \ncubic meter of deposit per thousand years")



# save plot  
jhe\_90mm\_ggsave("figures/fig\_4\_Jeremalai-flake-discard-phase-m3.png", height = 90)

# Results: Artefact taphonomy

The code chunk below computes a Bayesian Poisson exponential ANOVA to investigate differences in flake breakage classes over time.

allchert <- all[all$Material == 'Chert', ]  
allchert$phase <- makephases(allchert$Spit)  
# make Artclass that is long and transv breaks  
allchert$Artclas <- ifelse(allchert$Breaks == "",   
 as.character(allchert$Artclas),   
 paste(allchert$Artclas, allchert$Breaks, sep = "-"))  
taph <- data.frame(table(allchert$Artclas))  
# use regex to get broken flakes -b-   
broken <- allchert[grep("-b", allchert$Artclas), ]  
# get counts of broken to complete per phase  
# flake to -b-  
breaks <- dcast(allchert, Artclas ~ phase)[-1,]  
breaks <- breaks[breaks$Artclas =="Flake" | grepl("-b-", breaks$Artclas), ]  
allchert$Artclas <- tolower(allchert$Artclas)  
allchert$breakt <- "" # create variable to fill  
allchert$breakt[grep("trans", allchert$Artclas)] <- "trans"  
allchert$breakt[grep("long", allchert$Artclas)] <- "long"  
# per depositional phase  
breakt <- dcast(allchert, breakt ~ phase)[-1,]  
# add complete flake counts  
breakt <- rbind( breakt , setNames( breaks[1, ] , names( breakt ) ) )  
# shift rownames out and delete them  
rownames(breakt) <- breakt[,1]  
breakt <- breakt[,-1]  
# exclude artefacts not assigned to a phase  
breakt <- breakt[, -which(names(breakt) == "NA")]  
# do bayesian contingency table test  
colnames(breakt) <- c("phase 1", "phase 2", "phase 3", "phase 4")  
data <- melt(as.matrix(breakt), varnames=c("breakt", "phase"), value.name="Freq")  
  
myDataFrame = data  
yName=names(myDataFrame)[3] # Freq   
x1Name=names(myDataFrame)[2] # phase  
x2Name=names(myDataFrame)[1] # break type  
x1contrasts = list(   
 list( c("phase 1") , c("phase 2") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("phase 2") , c("phase 3") , compVal=0.0 , ROPE=c(-0.1,0.1)) ,  
 list( c("phase 3") , c("phase 4") , compVal=0.0 , ROPE=c(-0.1,0.1))   
 #list( c("phase 4") , c("phase 5") , compVal=0.0 , ROPE=c(-0.1,0.1))  
 )  
numSavedSteps = 12000 # MCMC parameters  
thinSteps = 10 # MCMC parameters  
  
mcmcCoda = genMCMC\_cont\_table( datFrm=myDataFrame ,   
 yName=yName , x1Name=x1Name , x2Name=x2Name ,  
 numSavedSteps=numSavedSteps , thinSteps=thinSteps ,   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 10:05:01 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 178  
  
WARNING: Unused variable(s) in data table:  
ySum  
  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 1000  
-------------------------------------------------| 1000  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 2000  
-------------------------------------------------| 2000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . . . . . . . Failed to set trace monitor for delta  
Node not found  
. Updating 40000  
-------------------------------------------------| 40000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . Updating 0  
. Deleting model  
.   
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
summaryInfo = smryMCMC\_cont\_table( mcmcCoda ,   
 datFrm=myDataFrame , x1Name=x1Name , x2Name=x2Name ,  
 x1contrasts=x1contrasts   
 )  
  
# # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_cont\_table( mcmcCoda ,   
# datFrm=myDataFrame , yName=yName , x1Name=x1Name ,   
# x2Name=x2Name ,  
# x1contrasts=x1contrasts )  
  
# table of raw counts  
kable(breakt, caption = "Table of frequencies of each class of breakage by phase")

Table of frequencies of each class of breakage by phase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | phase 1 | phase 2 | phase 3 | phase 4 |
| long | 58 | 31 | 63 | 57 |
| trans | 125 | 77 | 242 | 123 |
| Flake | 399 | 312 | 978 | 737 |

# subset summaryInfo to show HDI interval for interactions  
HDI\_intervals\_for\_interactions <- summaryInfo[c((nrow(summaryInfo)-length(x1contrasts)+1):nrow(summaryInfo)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions,3), caption = "Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake breakage classes.")

Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake breakage classes.

|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| phase 1.v.phase 2 | 0.268 | 0.616 |
| phase 2.v.phase 3 | -1.175 | -0.842 |
| phase 3.v.phase 4 | 0.224 | 0.498 |

The above code chunk returns results that there are credible difference but small differences in the frequences of flake breakage types by phase.

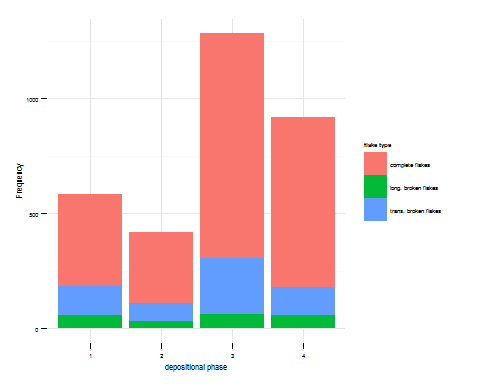
The code chunk below computes the frequentist equivalent, the chi-square test and the Cramer's V for effect size.

artefact\_taphonomy\_nhst <- assocstats(as.matrix(breakt))

The code chunk above returns a chi-squared value of 38.281632 and a p-value of 9.895611810^{-7}, indicating a signficant difference in frequencies of breakage classes by phase. However, the Cramer's V value of 0.077316 indicates that the effect size is extremely small. We interpret this result to mean that although the test result is statistically significant, there is no substantial significance in the differences in frequencies of breakage types by phase.

The code chunk below draws a plot of frequencies of complete flakes, transversely broken flakes and longitudinally broken flakes made from chert in each depositional unit at Jerimalai square B.

data$phase <- gsub("[[:alpha:]]\*", "", data$phase)  
# recode break type for pretty legend  
data$breakt\_ <- with(data, ifelse(breakt == 'long', 'long. broken flakes',  
 ifelse(breakt == "trans", "trans. broken flakes",  
 ifelse(breakt == "Flake", "complete flakes", NA))))  
ggplot(data, aes(phase, Freq, fill = breakt\_)) +  
 ylab("Frequency") +  
 geom\_bar(stat = "identity") +  
 scale\_fill\_discrete(name="flake type") +  
 xlab("depositional phase") +  
 theme\_minimal((base\_size = 6))



jhe\_90mm\_ggsave("figures/fig\_5\_Jeremalai-flake-broken-phase.png", height = 90/2)

The code chunk below summarises the frequencies of heat-treated flakes at Jerimalai square B

check <- sum(flakes$Heat, na.rm = TRUE) / nrow(flakes)  
heat <- aggregate(Heat ~ phase, flakes, length)  
total <- aggregate(Spit ~ phase, flakes, length)  
heat$Not\_heat <- total$Spit - heat$Heat  
# show proportions that are heat-treated  
max\_heat <- max(heat$Heat / total$Spit)  
min\_heat <- min(heat$Heat / total$Spit)  
heat <- t(heat)  
# do bayesian contingency table test  
colnames(heat) <- c("phase 1", "phase 2", "phase 3", "phase 4")  
heat <- heat[-1,]

Between 9% and 14% of chert artefacts in each depositional phase show signs of having been heated, such as crenation, potlid scars or surface crazing.

The code chunk below computes a Bayesian Poisson exponential ANOVA to investigate differences in heat treatment by phase.

data <- melt(as.matrix(heat), varnames=c("heat", "phase"), value.name="Freq")  
myDataFrame = data  
yName=names(myDataFrame)[3] # Freq   
x1Name=names(myDataFrame)[2] # phase  
x2Name=names(myDataFrame)[1] # heat  
x1contrasts = list(   
 list( c("phase 1") , c("phase 2") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("phase 2") , c("phase 3") , compVal=0.0 , ROPE=c(-0.1,0.1)) ,  
 list( c("phase 3") , c("phase 4") , compVal=0.0 , ROPE=c(-0.1,0.1))   
   
 )  
numSavedSteps = 12000 # MCMC parameters  
thinSteps = 10 # MCMC parameters  
  
mcmcCoda = genMCMC\_cont\_table( datFrm=myDataFrame ,   
 yName=yName , x1Name=x1Name , x2Name=x2Name ,  
 numSavedSteps=numSavedSteps , thinSteps=thinSteps ,   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 11:39:24 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 136  
  
WARNING: Unused variable(s) in data table:  
ySum  
  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 1000  
-------------------------------------------------| 1000  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 2000  
-------------------------------------------------| 2000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . . . . . . . Failed to set trace monitor for delta  
Node not found  
. Updating 40000  
-------------------------------------------------| 40000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . Updating 0  
. Deleting model  
.   
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
summaryInfo = smryMCMC\_cont\_table( mcmcCoda ,   
 datFrm=myDataFrame , x1Name=x1Name , x2Name=x2Name ,  
 x1contrasts=x1contrasts   
 )  
  
# # # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_cont\_table( mcmcCoda ,   
# datFrm=myDataFrame , yName=yName , x1Name=x1Name ,   
# x2Name=x2Name ,  
# x1contrasts=x1contrasts )  
  
# table of heated artefact counts  
kable(heat, caption = "Table of frequencies of heat treatment by phase")

Table of frequencies of heat treatment by phase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | phase 1 | phase 2 | phase 3 | phase 4 |
| Heat | 66 | 34 | 128 | 78 |
| Not\_heat | 409 | 337 | 1002 | 837 |

# subset summaryInfo to show HDI interval for interactions  
HDI\_intervals\_for\_interactions <- summaryInfo[c((nrow(summaryInfo)-length(x1contrasts)+1):nrow(summaryInfo)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions,3), caption = "Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake heat treatment.")

Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake heat treatment.

|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| phase 1.v.phase 2 | 0.180 | 0.611 |
| phase 2.v.phase 3 | -1.371 | -0.997 |
| phase 3.v.phase 4 | 0.179 | 0.468 |

The code chunk above returned results that indicate credible differences in the frequency of heat treatment between each phase except for phase one and two. The phase three to four transition is particularly different from the others, but this corresponds to only a 8.4% change increase in frequency of heated peices. all the HDIs are close to zero, indicating a small effect size.

The code chunk below computes a frequentist chi-square and Cramer's V test for the heat alteration data.

chert\_artefacts\_heat\_nhst <- assocstats(as.matrix(heat))

The code chunk above returns a chi-squared value of 11.0514076 and a p-value of 0.0114511, indicating a signficant difference in frequencies of breakage classes by phase. However, the Cramer's V value of 0.0618279 indicates that the effect size is extremely small. We interpret this result to mean that although the test result is statistically significant, there is no substantial significance in the differences in frequencies of heat alteration by phase.

# Results: Metric and technological characteristics of cores and unretouched flakes

The code chunk below produces the table "summarizing the attributes of chert complete flakes from Jerimalai square B. Each cell contains mean ± standard deviation unless otherwise indicated. The table produced here has been rearranged by hand for the paper.

# the table has been rearranged by hand for the paper  
metrics <- flakes %>%   
 group\_by(phase) %>%   
 summarise(median(Length, na.rm = TRUE),   
 median(Width, na.rm = TRUE),   
 median(Thick, na.rm = TRUE),  
 median(Weight, na.rm = TRUE),  
 median(Length, na.rm = TRUE),  
 median(Platwid, na.rm = TRUE),  
 median(Platthic, na.rm = TRUE),  
 median(NoDS, na.rm = TRUE),  
 median(Cortex, na.rm = TRUE),  
 IQR(Length, na.rm = TRUE),   
 IQR(Width, na.rm = TRUE),   
 IQR(Thick, na.rm = TRUE),  
 IQR(Weight, na.rm = TRUE),  
 IQR(Length, na.rm = TRUE),  
 IQR(Platwid, na.rm = TRUE),  
 IQR(Platthic, na.rm = TRUE),  
 IQR(NoDS, na.rm = TRUE),  
 IQR(Cortex, na.rm = TRUE),  
 n = length(Weight))  
  
# get overhang removal data also  
ohr <- filter(flakes, Overhang == "Yes") %>%  
 group\_by(phase) %>%  
 summarise(OHR\_n = length(Overhang))  
# get percentages of OHR per phase  
ohr$OHR\_perc <- ohr$OHR\_n/metrics$n \* 100  
  
# combine  
metrics <- cbind(metrics, ohr[,c("OHR\_n", "OHR\_perc")])  
metrics <- t(round(metrics,2))  
# save as CSV to tidy up, there is no simple way to make the table  
# that appears in the paper  
write.csv(metrics, "figures/table\_6\_flake\_metrics.csv")  
# show a slightly untidy version  
metrics <- data.frame(metrics)  
names(metrics) <- c("phase 1", "phase 2", "phase 3", "phase 4")  
kable(metrics[-1,], caption = "Summary of attributes of chert complete flakes from Jerimalai square B. Each cell contains median ± interquartile range unless otherwise indicated. ")

Summary of attributes of chert complete flakes from Jerimalai square B. Each cell contains median ± interquartile range unless otherwise indicated.

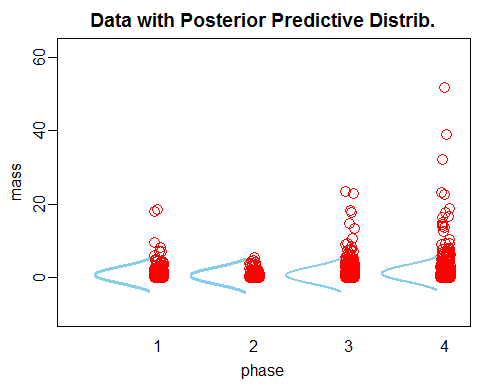
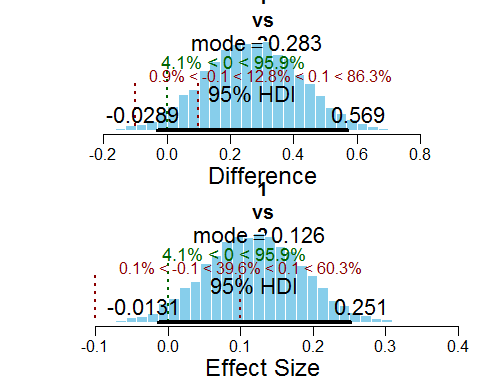
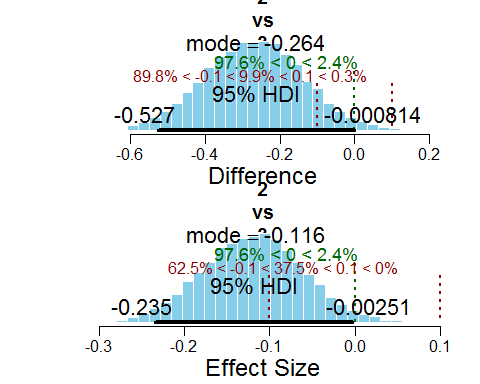
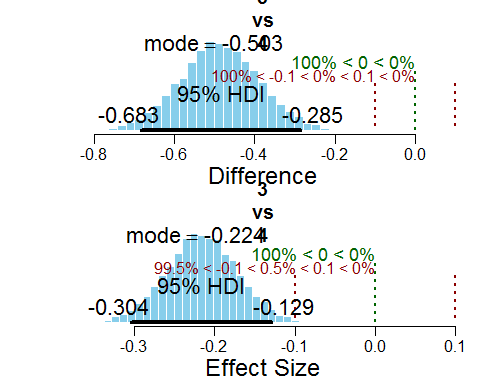
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | phase 1 | phase 2 | phase 3 | phase 4 | NA |
| median(Length, na.rm = TRUE) | 9.00 | 7.99 | 9.22 | 10.44 | 6.46 |
| median(Width, na.rm = TRUE) | 8.47 | 7.78 | 8.50 | 9.71 | 6.58 |
| median(Thick, na.rm = TRUE) | 2.15 | 1.84 | 2.21 | 2.53 | 2.09 |
| median(Weight, na.rm = TRUE) | 0.18 | 0.12 | 0.19 | 0.31 | 0.11 |
| median(Platwid, na.rm = TRUE) | 6.16 | 5.92 | 6.46 | 7.18 | 5.84 |
| median(Platthic, na.rm = TRUE) | 2.21 | 1.68 | 2.16 | 2.53 | 1.87 |
| median(NoDS, na.rm = TRUE) | 4.00 | 4.00 | 4.00 | 4.00 | 3.00 |
| median(Cortex, na.rm = TRUE) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| IQR(Length, na.rm = TRUE) | 7.81 | 6.61 | 6.78 | 8.95 | 4.21 |
| IQR(Width, na.rm = TRUE) | 6.87 | 5.75 | 6.79 | 7.96 | 5.25 |
| IQR(Thick, na.rm = TRUE) | 2.70 | 2.02 | 2.37 | 3.06 | 1.18 |
| IQR(Weight, na.rm = TRUE) | 0.55 | 0.37 | 0.55 | 0.91 | 0.19 |
| IQR(Platwid, na.rm = TRUE) | 5.88 | 4.80 | 5.35 | 5.67 | 3.22 |
| IQR(Platthic, na.rm = TRUE) | 2.48 | 1.81 | 2.08 | 2.67 | 1.32 |
| IQR(NoDS, na.rm = TRUE) | 2.00 | 2.00 | 2.00 | 3.00 | 2.00 |
| IQR(Cortex, na.rm = TRUE) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| n | 475.00 | 371.00 | 1130.00 | 915.00 | 51.00 |
| OHR\_n | 259.00 | 198.00 | 651.00 | 542.00 | 33.00 |
| OHR\_perc | 54.53 | 53.37 | 57.61 | 59.23 | 64.71 |

The code chunk below uses a Bayesian one-way ANOVA to investigate differences in chert flake mass by depositional phase.

myDataFrame <- data.frame(phase = flakes$phase, mass = flakes$Weight)  
# remove flakes with no phase  
myDataFrame <- myDataFrame[!is.na(myDataFrame$phase),]  
yName = names(myDataFrame)[2] # mass  
xName = names(myDataFrame)[1] # phase  
contrasts = list(  
 list( c("1") , c("2") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("2") , c("3") , compVal=0.0 , ROPE=c(-0.1,0.1)) ,  
 list( c("3") , c("4") , compVal=0.0 , ROPE=c(-0.1,0.1)) )  
# Generate the MCMC chain:  
 mcmcCoda\_ANOVA = genMCMC\_ANOVA(datFrm=myDataFrame , yName=yName , xName=xName ,  
 numSavedSteps=11000 , thinSteps=10   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 10:05:27 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 5820  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 500  
-------------------------------------------------| 500  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 1000  
-------------------------------------------------| 1000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . Updating 36670  
-------------------------------------------------| 36650  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
\* 100%  
. . . . Updating 0  
. Deleting model  
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
 summaryInfo\_ANOVA = smryMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
 datFrm=myDataFrame , xName=xName ,  
 contrasts=contrasts   
 )  
   
# Display posterior information (not easy to read unless using interactively):  
 plotMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
 datFrm=myDataFrame , yName=yName , xName=xName ,   
 contrasts=contrasts )

# subset summaryInfo to show HDI interval for interactions  
 HDI\_intervals\_for\_interactions <- summaryInfo\_ANOVA[c((nrow(summaryInfo\_ANOVA)-length(contrasts)+1):nrow(summaryInfo\_ANOVA)), c("HDIlow", "HDIhigh")]  
  
kable(round(HDI\_intervals\_for\_interactions,3), caption = "Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake mass.")

Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake mass.

|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| 1.v.2 | -0.029 | 0.569 |
| 2.v.3 | -0.527 | -0.001 |
| 3.v.4 | -0.683 | -0.285 |

The code chunk above produced results that indicate that only phase four and five have a credibly different distributions of flake mass. In the code chunk below we repeat the same investigation using a frequentist ANOVA.

# ANOVA with Tukey's HSD  
fit <- aov(Weight ~ as.factor(phase), flakes)  
fit\_summary <- summary(fit)  
tuk <- TukeyHSD(fit)  
# plot(tuk, las = 2, cex = 0.1)  
kable(round(tuk$`as.factor(phase)`,3), caption = "Tukey's Honest Significant Difference for phase by phase comparisons of flake mass.")

Tukey's Honest Significant Difference for phase by phase comparisons of flake mass.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | diff | lwr | upr | p adj |
| 2-1 | -0.291 | -0.694 | 0.111 | 0.245 |
| 3-1 | 0.004 | -0.314 | 0.321 | 1.000 |
| 4-1 | 0.508 | 0.180 | 0.836 | 0.000 |
| 3-2 | 0.295 | -0.052 | 0.642 | 0.128 |
| 4-2 | 0.799 | 0.442 | 1.157 | 0.000 |
| 4-3 | 0.504 | 0.246 | 0.762 | 0.000 |

In the code chunk above we obtain the results of a frequentist ANOVA which returns a statistically significant result: F = 14.5396391, df = 3, p = 2.125079410^{-9}. The Tukey's Honest Significant Difference test shows confidence intervals on the differences between the means artefact masses of a phase. When the interval includes zero, the difference is considered not significant. In this case we see that phase five differs from phase four, equivalent to what we see in the Bayesian HDI intervals.

The code chunk below produces a summary table of metric attributes of chert complete cores from Jerimalai square B.

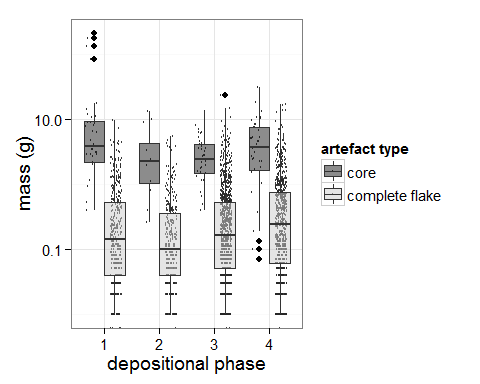
cores\_mass <- allchert[allchert$Artclas == "core", ]  
core\_metrics <- cores\_mass %>%   
 group\_by(phase) %>%   
 filter(phase %in% 1:4) %>%  
 summarise(median(Weight),   
 median(Length),   
 median(Width),   
 median(Thick),  
 median(as.numeric(cores\_mass$NoDS)),  
 median(as.numeric(cores\_mass$Cortex), na.rm = TRUE),  
 median(as.numeric(cores\_mass$Corerot)),  
 IQR(Weight),   
 IQR(Length),   
 IQR(Width),   
 IQR(Thick),  
 IQR(as.numeric(cores\_mass$NoDS)),  
 IQR(as.numeric(cores\_mass$Cortex), na.rm = TRUE),  
 IQR(as.numeric(cores\_mass$Corerot)),  
 n = length(Weight))  
core\_metrics\_t <- t(round(core\_metrics,2))  
colnames(core\_metrics\_t) <- c("phase 1", "phase 2", "phase 3", "phase 4" )  
kable(core\_metrics\_t, caption = "Summary of chert core metrics")

Summary of chert core metrics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | phase 1 | phase 2 | phase 3 | phase 4 |
| phase | 1.00 | 2.00 | 3.00 | 4.00 |
| median(Weight) | 3.79 | 2.25 | 2.38 | 3.65 |
| median(Length) | 22.94 | 17.82 | 17.45 | 20.21 |
| median(Width) | 15.28 | 14.66 | 13.55 | 14.80 |
| median(Thick) | 10.62 | 6.75 | 8.78 | 9.56 |
| median(as.numeric(cores\_mass$NoDS)) | 11.00 | 11.00 | 11.00 | 11.00 |
| median(as.numeric(cores\_mass$Cortex),... | 0.00 | 0.00 | 0.00 | 0.00 |
| median(as.numeric(cores\_mass$Corerot)) | 3.50 | 3.50 | 3.50 | 3.50 |
| IQR(Weight) | 7.05 | 3.70 | 2.67 | 5.83 |
| IQR(Length) | 13.46 | 8.28 | 8.66 | 11.13 |
| IQR(Width) | 10.07 | 9.59 | 5.37 | 9.18 |
| IQR(Thick) | 6.87 | 4.21 | 4.84 | 3.44 |
| IQR(as.numeric(cores\_mass$NoDS)) | 5.00 | 5.00 | 5.00 | 5.00 |
| IQR(as.numeric(cores\_mass$Cortex), na... | 5.00 | 5.00 | 5.00 | 5.00 |
| IQR(as.numeric(cores\_mass$Corerot)) | 1.00 | 1.00 | 1.00 | 1.00 |
| n | 36.00 | 12.00 | 37.00 | 41.00 |

The code chunk below creates a figure that shows the distribution of core and complete flake mass by depositional phases

# plot flake and core mass by phase in one box plot  
flakes\_cores\_weight <- allchert %>%  
 filter(Artclas %in% c("flake", "core")) %>%  
 filter(phase %in% 1:4) %>%  
 select(Artclas, Weight, phase)  
   
  
ggplot(flakes\_cores\_weight, aes(fill = Artclas, as.factor(phase), Weight)) +  
 geom\_point(aes(colour = Artclas), size = 0.5, alpha = 0.9, shape = 1,   
 position=position\_jitterdodge(dodge.width=0.9)) +  
 geom\_boxplot(alpha = 0.5) +  
 scale\_y\_log10() +  
 #theme\_minimal(base\_size = 4) +  
 xlab('depositional phase') +  
 ylab("mass (g)") +  
 scale\_fill\_manual(name="artefact type",  
 labels=c("core", "complete flake"),   
 values = c("grey10", "grey80")) +  
 scale\_colour\_manual(name="artefact type",  
 labels=c("core", "complete flake"),   
 values = c("grey10", "grey15"))



# save plot  
jhe\_190mm\_ggsave("figures/fig\_6\_Jeremalai-flake-core-mass-phase.png",   
 height = (190/2))

The code chunk below uses a Bayesian one-way ANOVA to investigate differences in chert core mass by depositional phase.

myDataFrame <- data.frame(phase = cores\_mass$phase, mass = cores\_mass$Weight)  
# remove flakes with no phase  
myDataFrame <- myDataFrame[!is.na(myDataFrame$phase),]  
yName = names(myDataFrame)[2] # mass  
xName = names(myDataFrame)[1] # phase  
contrasts = list(  
 list( c("1") , c("2") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("2") , c("3") , compVal=0.0 , ROPE=c(-0.1,0.1)) ,  
 list( c("3") , c("4") , compVal=0.0 , ROPE=c(-0.1,0.1)) )  
# Generate the MCMC chain:  
 mcmcCoda\_ANOVA = genMCMC\_ANOVA(datFrm=myDataFrame , yName=yName , xName=xName ,  
 numSavedSteps=11000 , thinSteps=10   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 10:07:10 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 290  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 500  
-------------------------------------------------| 500  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 1000  
-------------------------------------------------| 1000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . Updating 36670  
-------------------------------------------------| 36650  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
\* 100%  
. . . . Updating 0  
. Deleting model  
.   
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
 summaryInfo\_ANOVA = smryMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
 datFrm=myDataFrame , xName=xName ,  
 contrasts=contrasts   
 )  
   
# # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
# datFrm=myDataFrame , yName=yName , xName=xName ,   
# contrasts=contrasts )  
#   
# subset summaryInfo to show HDI interval for interactions  
 HDI\_intervals\_for\_interactions <- summaryInfo\_ANOVA[c((nrow(summaryInfo\_ANOVA)-length(contrasts)+1):nrow(summaryInfo\_ANOVA)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions,3), caption = "Highest density intervals (95%) for the posterior distributions of the interactions between phases and core mass.")

Highest density intervals (95%) for the posterior distributions of the interactions between phases and core mass.

|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| 1.v.2 | -1.927 | 29.103 |
| 2.v.3 | -13.307 | 15.800 |
| 3.v.4 | -12.546 | 9.162 |

The code chunk above produced results that indicate that the HDIs for all phases include zero, which we interpret as no credible difference. In the code chunk below we repeat the same investigation using a frequentist ANOVA.

# ANOVA with Tukey's HSD  
fit <- aov(mass ~ as.factor(phase), myDataFrame)  
fit\_summary <- summary(fit)  
tuk <- TukeyHSD(fit)  
# plot(tuk, las = 2, cex = 0.1)  
kable(round(tuk$`as.factor(phase)`,3), caption = "Tukey's Honest Significant Difference for phase by phase comparisons of core mass.")

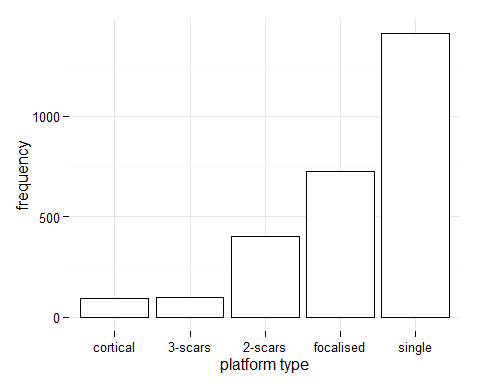
Tukey's Honest Significant Difference for phase by phase comparisons of core mass.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | diff | lwr | upr | p adj |
| 2-1 | -17.283 | -40.336 | 5.771 | 0.212 |
| 3-1 | -17.997 | -34.187 | -1.806 | 0.023 |
| 4-1 | -15.426 | -31.223 | 0.371 | 0.058 |
| 3-2 | -0.714 | -23.690 | 22.262 | 1.000 |
| 4-2 | 1.856 | -20.843 | 24.556 | 0.997 |
| 4-3 | 2.570 | -13.112 | 18.253 | 0.974 |

In the code chunk above we obtain the results of a frequentist ANOVA which returns a statistically significant result: F = 3.4659527, df = 3, p = 0.0184061. The Tukey's Honest Significant Difference test shows confidence intervals on the differences between the means artefact masses of a phase. In this case we see that only the comparison of phase four to phase two has a significant difference in core mass. Consequtive phases show no signifance difference, which we interpret as evidence of overall no substantial change in core mass.

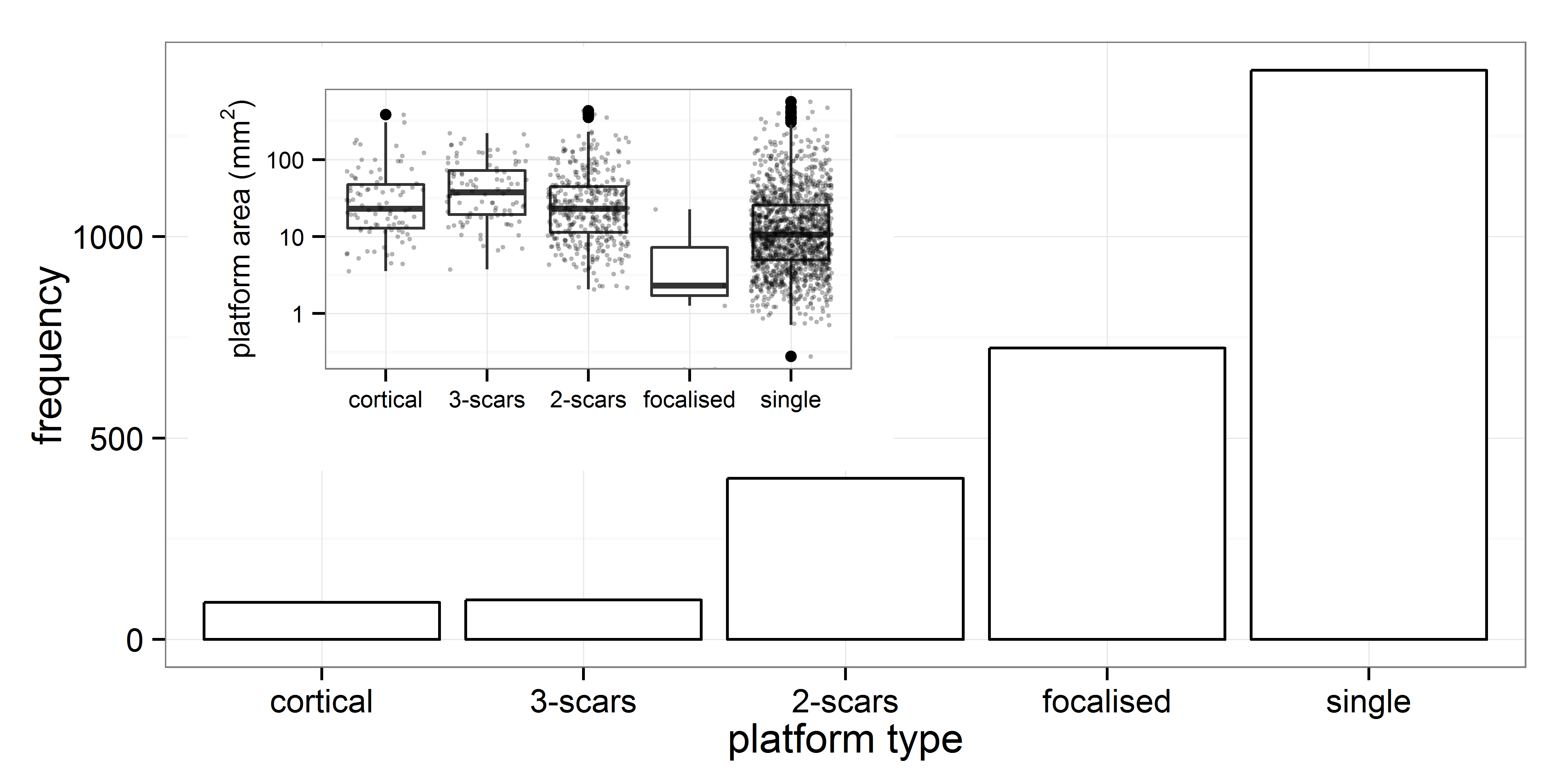
The code chunk below produces the figures that illustrate the frequency of flake platform categories for chert complete flakes at Jerimalai square B.

# flake platform  
plat <- dcast(flakes, Plat ~ depth)   
rownames(plat) <- plat[,1]  
# get rid of rows with no plat  
plat <- plat[rownames(plat) != "",]  
plat <- plat[,-1]  
plat\_freqs <- data.frame(plat\_types = rownames(plat), Freq = rowSums(plat))  
plat\_freqs <- plat\_freqs[plat\_freqs$Freq > 90,]  
plat\_freqs$plat\_types <- c('2-scars', '3-scars', 'cortical', 'focalised', 'single')  
# plot freq of platform types  
main <- ggplot(plat\_freqs, aes(reorder(plat\_types, Freq), Freq)) +   
 geom\_bar(stat="identity", fill = "white", colour = "black") +  
 # theme\_minimal() +  
 xlab("platform type") +  
 ylab("frequency") +  
 # remove grid lines for subplot  
 theme\_update(panel.background = element\_blank(),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank())  
main + theme\_minimal()



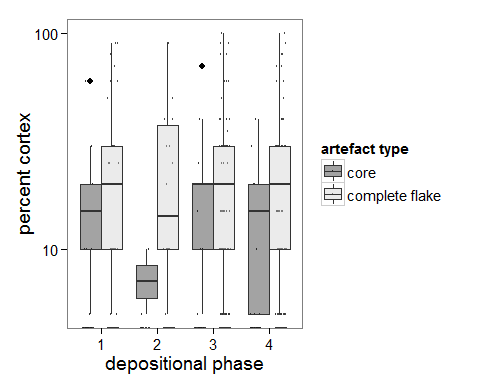
# raw materials by spit  
# compute proportions per layer (col props)  
all\_tab <- data.frame()  
for(i in seq(ncol(plat))){  
 for(j in seq(nrow(plat))){  
 all\_tab[j,i] <- plat[j,i]/colSums(plat)[i]  
 }  
}  
# check   
check <- colSums(all\_tab) # should == 1  
colnames(all\_tab) <- colnames(plat)   
check <- rowSums(all\_tab) # should be various  
all\_tab$plat\_type <- rownames(plat)   
# get rid of raw materials that are not very abundant  
all\_tab <- all\_tab[which(rowSums(all\_tab[,1:ncol(all\_tab)-1]) > 1.5) , ]  
# get rid of NA column  
all\_tab <- all\_tab[,names(all\_tab) != 'NA']  
# plot  
all\_tab\_m <- melt(all\_tab, id.var = 'plat\_type')  
# by phase  
plat <- dcast(flakes, Plat ~ phase)  
rownames(plat) <- plat[,1]  
# get rid of rows with no plat  
plat <- plat[rownames(plat) != "",]  
plat <- plat[,-1]  
# raw materials by site  
# compute proportions per phase (col props)  
all\_tab <- data.frame()  
for(i in seq(ncol(plat))){  
 for(j in seq(nrow(plat))){  
 all\_tab[j,i] <- plat[j,i]/colSums(plat)[i]  
 }  
}  
all\_tab$plat\_type <- rownames(plat)   
# get rid of raw materials that are not very abundant  
all\_tab <- all\_tab[which(rowSums(all\_tab[,1:ncol(all\_tab)-1]) > 0.15) , ]  
# get rid of NA column  
all\_tab <- all\_tab[,names(all\_tab) != 'NA']  
# plot distibution of platform sizes for each type  
flakes$Platarea <- with(flakes, (Platthic \* Platwid))  
plat\_area\_type <- flakes[flakes$Plat %in% c("Single", "Focal", "2-scars", "Cort", '3-scars'),]  
  
# make names a bit more readable  
plat\_area\_type$Plat <- ifelse(plat\_area\_type$Plat == 'Cort', 'cortical',  
 ifelse(plat\_area\_type$Plat == 'Focal', 'focalised',   
 ifelse(plat\_area\_type$Plat == 'Single', 'single', as.character(plat\_area\_type$Plat))))   
  
# put types in same order as frequency plot  
plat\_area\_type$Plat <- factor(plat\_area\_type$Plat,   
 levels = c('cortical', '3-scars','2-scars', 'focalised', 'single'), ordered = TRUE)  
   
sub <- ggplot(plat\_area\_type, aes(Plat, Platarea)) +  
 geom\_boxplot() +  
 geom\_jitter(size = 0.5, alpha = 0.3, shape = 1) +  
 scale\_y\_log10() +  
 ylab(as.expression(bquote('platform area (' \* mm^{2} \* ")" ))) +  
 xlab("") +  
 theme\_minimal()  
  
# plot freq of plat type and platform area together in one plot  
   
vp <- viewport(width = 0.45, height = 0.54,   
 x = 0.57, y = 0.4,   
 just = c("right", "bottom"))  
  
# combine plots, print and save (wont show in console)  
 png("figures/fig\_7\_Jeremalai-platform-area-by-plat-type.png",   
 units = "mm", w = 190, h = 190/2, res = 600)  
 print(main)  
 print(sub + theme\_bw(base\_size = 10), vp = vp)  
 dev.off()

png   
 2



The code chunk below creates a plot showing the distributions of core and flake cortex by depositional phase at Jerimalai square B.

flakes\_cores\_cortex <- allchert %>%  
 filter(Artclas %in% c("flake", "core")) %>%  
 filter(phase %in% 1:4) %>%  
 select(Artclas, Cortex, phase)  
  
ggplot(flakes\_cores\_cortex, aes( fill = Artclas, as.factor(phase), Cortex)) +  
 geom\_point(aes(colour = Artclas), size = 0.5, alpha = 0.9, shape = 1,   
 position=position\_jitterdodge(dodge.width=0.9)) +  
 geom\_boxplot(alpha = 0.4) +  
 scale\_y\_log10() +  
 xlab('depositional phase') +  
 ylab("percent cortex") +  
 scale\_fill\_manual(name="artefact type",  
 labels=c("core", "complete flake"),   
 values = c("grey10", "grey80")) +  
 scale\_colour\_manual(name="artefact type",  
 labels=c("core", "complete flake"),   
 values = c("grey10", "grey20"))



# save plot  
jhe\_190mm\_ggsave("figures/fig\_8\_Jeremalai-flake-core-cortex-phase.png", height = 190/2)

The code chunk below uses a Bayesian one-way ANOVA to investigate differences in chert flake cortex by depositional phase.

myDataFrame <- data.frame(phase = flakes$phase, cortex = flakes$Cortex)  
# remove flakes with no phase  
myDataFrame <- myDataFrame[!is.na(myDataFrame$phase),]  
yName = names(myDataFrame)[2] # cortex  
xName = names(myDataFrame)[1] # phase  
contrasts = list(  
 list( c("1") , c("2") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("2") , c("3") , compVal=0.0 , ROPE=c(-0.1,0.1)) ,  
 list( c("3") , c("4") , compVal=0.0 , ROPE=c(-0.1,0.1)) )  
# Generate the MCMC chain:  
 mcmcCoda\_ANOVA = genMCMC\_ANOVA(datFrm=myDataFrame , yName=yName , xName=xName ,  
 numSavedSteps=11000 , thinSteps=10   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 10:07:24 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 5820  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 500  
-------------------------------------------------| 500  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 1000  
-------------------------------------------------| 1000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . Updating 36670  
-------------------------------------------------| 36650  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
\* 100%  
. . . . Updating 0  
. Deleting model  
.   
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
 summaryInfo\_ANOVA = smryMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
 datFrm=myDataFrame , xName=xName ,  
 contrasts=contrasts   
 )  
 # # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
# datFrm=myDataFrame , yName=yName , xName=xName ,   
# contrasts=contrasts )  
#   
# subset summaryInfo to show HDI interval for interactions  
HDI\_intervals\_for\_interactions\_f <- summaryInfo\_ANOVA[c((nrow(summaryInfo\_ANOVA)-length(contrasts)+1):nrow(summaryInfo\_ANOVA)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions\_f,3), caption = "Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake cortex")

Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake cortex

|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| 1.v.2 | -0.743 | 2.192 |
| 2.v.3 | -1.601 | 1.002 |
| 3.v.4 | -2.058 | 0.135 |

The code chunk above produced results that indicate that the HDIs for all phases include zero, which we interpret as no credible difference. In the code chunk below we repeat the same investigation using a frequentist ANOVA.

# ANOVA with Tukey's HSD  
fit <- aov(cortex ~ as.factor(phase), myDataFrame)  
fit\_summary <- summary(fit)  
tuk <- TukeyHSD(fit)  
# plot(tuk, las = 2, cex = 0.1)  
kable(round(tuk$`as.factor(phase)`,3), caption = "Tukey's Honest Significant Difference for phase by phase comparisons of flake cortex")

Tukey's Honest Significant Difference for phase by phase comparisons of flake cortex

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | diff | lwr | upr | p adj |
| 2-1 | -0.940 | -3.228 | 1.348 | 0.716 |
| 3-1 | -0.491 | -2.297 | 1.314 | 0.897 |
| 4-1 | 0.729 | -1.139 | 2.596 | 0.748 |
| 3-2 | 0.449 | -1.527 | 2.424 | 0.937 |
| 4-2 | 1.668 | -0.364 | 3.701 | 0.150 |
| 4-3 | 1.220 | -0.249 | 2.688 | 0.142 |

In the code chunk above we obtain the results of a frequentist ANOVA which returns a statistically nonsignificant result: F = 2.1550804, df = 3, p = 0.0912883. The Tukey's Honest Significant Difference test shows confidence intervals on the differences between the means artefact masses of a phase. None of the interactions show a significant difference. We interpret this as evidence of overall no change in flake cortex.

In the two code chunks below we repeat the statistical tests above for core cortex.

myDataFrame <- data.frame(phase = cores\_mass$phase, cortex = cores\_mass$Cortex)  
myDataFrame <- myDataFrame[complete.cases(myDataFrame),] # omit NAs  
yName = names(myDataFrame)[2] # cortex  
xName = names(myDataFrame)[1] # phase  
contrasts = list(  
 list( c("1") , c("2") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("2") , c("3") , compVal=0.0 , ROPE=c(-0.1,0.1)) ,  
 list( c("3") , c("4") , compVal=0.0 , ROPE=c(-0.1,0.1))   
 )  
# Generate the MCMC chain:  
 mcmcCoda\_ANOVA = genMCMC\_ANOVA(datFrm=myDataFrame , yName=yName , xName=xName ,  
 numSavedSteps=11000 , thinSteps=10   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 11:43:00 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 286  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 500  
-------------------------------------------------| 500  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 1000  
-------------------------------------------------| 1000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . Updating 36670  
-------------------------------------------------| 36650  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
\* 100%  
. . . . Updating 0  
. Deleting model  
.   
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
 summaryInfo\_ANOVA = smryMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
 datFrm=myDataFrame , xName=xName ,  
 contrasts=contrasts   
 )  
 # # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
# datFrm=myDataFrame , yName=yName , xName=xName ,   
# contrasts=contrasts )  
#   
# subset summaryInfo to show HDI interval for interactions  
 HDI\_intervals\_for\_interactions\_c <- summaryInfo\_ANOVA[c((nrow(summaryInfo\_ANOVA)-length(contrasts)+1):nrow(summaryInfo\_ANOVA)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions\_c,3), caption = "Highest density intervals (95%) for the posterior distributions of the interactions between phases and core cortex")

Highest density intervals (95%) for the posterior distributions of the interactions between phases and core cortex

|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| 1.v.2 | -3.094 | 8.893 |
| 2.v.3 | -11.194 | 1.797 |
| 3.v.4 | -1.523 | 7.973 |

# combine core and flake HDIs into one table  
HDIs\_for\_cores\_and\_flakes <- round(cbind(HDI\_intervals\_for\_interactions\_c, HDI\_intervals\_for\_interactions\_f),3)  
kable(HDIs\_for\_cores\_and\_flakes, caption = "Highest density intervals (95%) for the posterior distributions of the interactions between phases and core and flake cortex")

Highest density intervals (95%) for the posterior distributions of the interactions between phases and core and flake cortex

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | HDIlow | HDIhigh | HDIlow | HDIhigh |
| 1.v.2 | -3.094 | 8.893 | -0.743 | 2.192 |
| 2.v.3 | -11.194 | 1.797 | -1.601 | 1.002 |
| 3.v.4 | -1.523 | 7.973 | -2.058 | 0.135 |

The code chunk above produced results that indicate that the HDIs for all phases include zero, which we interpret as no credible difference. In the code chunk below we repeat the same investigation using a frequentist ANOVA.

# ANOVA with Tukey's HSD  
fit <- aov(cortex ~ as.factor(phase), myDataFrame)  
fit\_summary <- summary(fit)  
tuk <- TukeyHSD(fit)  
# plot(tuk, las = 2, cex = 0.1)  
kable(round(tuk$`as.factor(phase)`,3), caption = "Tukey's Honest Significant Difference for phase by phase comparisons of flake cortex")

Tukey's Honest Significant Difference for phase by phase comparisons of flake cortex

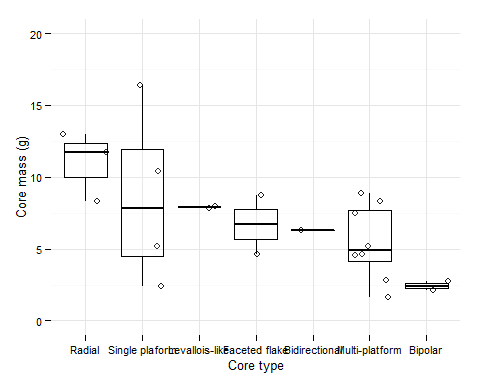
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | diff | lwr | upr | p adj |
| 2-1 | -4.583 | -14.548 | 5.381 | 0.629 |
| 3-1 | 2.680 | -4.318 | 9.678 | 0.751 |
| 4-1 | -1.987 | -8.896 | 4.922 | 0.877 |
| 3-2 | 7.264 | -2.667 | 17.194 | 0.231 |
| 4-2 | 2.596 | -7.272 | 12.464 | 0.902 |
| 4-3 | -4.667 | -11.528 | 2.193 | 0.292 |

In the code chunk above we obtain the results of a frequentist ANOVA which returns a statistically nonsignificant result: F = 1.6785207, df = 3, p = 0.1752574. The Tukey's Honest Significant Difference test shows confidence intervals on the differences between the means artefact masses of a phase. None of the interactions show a significant difference. We interpret this as evidence of overall no change in core cortex.

# Results: Core technology

This code draws a plot of differences in the mass of cores recovered from Jerimalai by core type.

sq\_b\_core\_types <- core\_types %>% filter(Square == "B")  
sq\_b\_core\_types$Type\_long <- with(sq\_b\_core\_types, ifelse(Type == "SPC", "Single plaform",  
 ifelse(Type == "RC", "Radial",  
 ifelse(Type == "BDC", "Bidirectional",  
 ifelse(Type == "BiC", "Bipolar",  
 ifelse(Type == "MPC", "Multi-platform",  
 ifelse(Type == "LLC", "Levallois-like",  
 ifelse(Type == "FFC", "Faceted flake", NA))))))) )  
# plot  
ggplot(sq\_b\_core\_types, aes(reorder(Type\_long, -Mass, FUN=median), Mass)) +  
 geom\_jitter(alpha = 0.9, shape = 1) +   
 geom\_boxplot(alpha = 0.1, fill = "white", colour = "black") +  
 ylim(0,20) +  
 xlab("Core type") +  
 ylab("Core mass (g)") +  
 theme\_minimal(base\_size = 10)



# save  
jhe\_190mm\_ggsave("figures/fig\_9\_Jeremalai-core-by-type.png", height = 190/1.6)

The code chunk below computes the amount of cortex for each core type.

core\_cortex <- aggregate( X..Cortex ~ Type, data = sq\_b\_core\_types, mean)  
# what is the average amount of cortex for each core type?  
core\_cortex\_means <- arrange(core\_cortex, -X..Cortex)  
names(core\_cortex\_means) <- c("Type", "Cortex percentage")  
core\_cortex\_means[,2] <- round(core\_cortex\_means[,2],1)

Consistent with overall size, single platform cores retain the most cortex on average (20%), followed by radial cores (10%), multiplatform cores (2.2%) and faceted flake cores (0%). Levallois-like cores and bipolar cores exhibit the least cortex (<4%).

The code chunk below computes the numbers of flake scars by each core type.

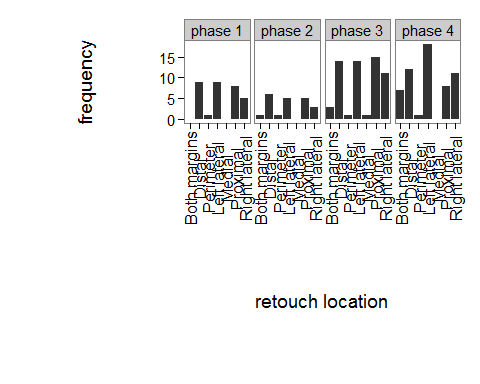
core\_scars <- sq\_b\_core\_types %>%  
 group\_by(Type) %>%  
 summarize(means = round(mean(Number.of.Scars, na.rm = TRUE),0),  
 sds = round(sd(Number.of.Scars, na.rm = TRUE)),0) %>%  
 arrange(means)

Levallois-like cores exhibit almost twice the number of flake scars on average as other cores in the assemblage (mean = 22±3 scars versus 15±3 on average). Single platform cores have the least scars on average (10±3).

# Results: Retouched artefacts

The code chunk below draws a plot showing locations of retouch on chert flakes by depositional unit at Jerimalai square B.

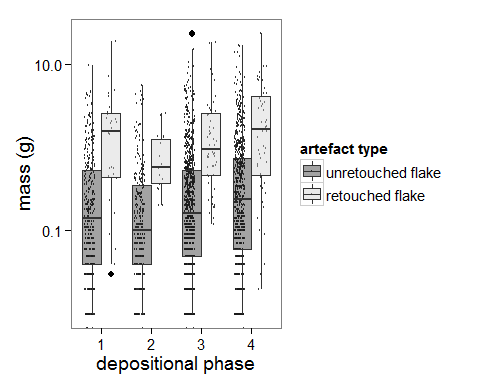
# frequency of flakes with retouch per phase  
rt <- flakes[flakes$Rtch == "Yes", ]  
rt\_tab <- t(data.frame(rt = aggregate(Retype ~ phase, rt, function(x) rt\_count = length(x)), flakes = aggregate(Retype ~ phase, flakes, function(x) fk\_count = length(x))[,2] ))  
  
# retouch locations  
rt\_loc <- data.frame(phase = rt$phase, rt\_loc = rt$Retloc)  
rt\_loc <- rt\_loc[!(is.na(rt\_loc$phase)), ]  
rt\_sum <- as.data.frame.matrix(table(rt\_loc))[,-1]  
rt\_sum$phase <- paste0("phase ", row.names(rt\_sum))  
colnames(rt\_sum) <- c("Both margins", "Distal", "Perimeter", "Left lateral", "Medial", "Proximal", "Right lateral", "phase")  
rt\_sum\_m <- melt(rt\_sum)  
# plot  
ggplot(rt\_sum\_m, aes(variable, value)) +  
 geom\_bar(stat="identity") +  
 theme(axis.text.x = element\_text(angle=90, vjust=0.5, size=12)) +  
 ylab("frequency") +  
 xlab("retouch location") +  
 facet\_wrap(~phase, ncol = 5)



# save plot  
jhe\_190mm\_ggsave("figures/fig\_15\_Jeremalai-flake-retouched-flake-location-phase.png", height = 190/1.6)

The code chunk below draws a plot of the distribution of flake lengths for retouched and unretouched flakes by depositional unit at Jerimalai square B.

flakes\_retouch\_size <- allchert %>%  
 filter(Artclas %in% c("flake", "retf")) %>%  
 filter(phase %in% 1:4) %>%  
 select(Artclas, Weight, Length, phase)  
  
ggplot(flakes\_retouch\_size, aes( fill = Artclas, as.factor(phase), Weight)) +  
 geom\_point(aes(colour = Artclas), size = 0.5, alpha = 0.9, shape = 1,   
 position=position\_jitterdodge(dodge.width=0.9)) +   
 geom\_boxplot(alpha = 0.4) +  
 scale\_y\_log10() +  
 xlab('depositional phase') +  
 ylab("mass (g)") +  
 scale\_fill\_manual(name="artefact type",  
 labels=c("unretouched flake", "retouched flake"),   
 values = c("grey10", "grey80")) +  
 scale\_colour\_manual(name="artefact type",  
 labels=c("unretouched flake", "retouched flake"),   
 values = c("grey10", "grey15"))



# save plot  
jhe\_190mm\_ggsave("figures/fig\_16\_Jeremalai-flake-retouchedflake-mass-phase.png", height = 190/1.6)

The code chunk below computes the tests for credible difference in retouch flake frequency by phase.

rt\_tab <- data.frame(rt\_tab)  
colnames(rt\_tab) <- c("phase 1", "phase 2", "phase 3", "phase 4")  
rt\_tab <- rt\_tab[-1,]  
myDataFrame <- melt(as.matrix(rt\_tab), varnames=c("retouch", "phase"), value.name="Freq")  
myDataFrame <- myDataFrame[complete.cases(myDataFrame),] # omit NAs  
yName=names(myDataFrame)[3] # Freq   
x1Name=names(myDataFrame)[2] # phase  
x2Name=names(myDataFrame)[1] # retouch  
x1contrasts = list(   
 list( c("phase 1") , c("phase 2") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("phase 2") , c("phase 3") , compVal=0.0 , ROPE=c(-0.1,0.1)) ,  
 list( c("phase 3") , c("phase 4") , compVal=0.0 , ROPE=c(-0.1,0.1))   
 )  
numSavedSteps = 12000 # MCMC parameters  
thinSteps = 10 # MCMC parameters  
  
mcmcCoda = genMCMC\_cont\_table( datFrm=myDataFrame ,   
 yName=yName , x1Name=x1Name , x2Name=x2Name ,  
 numSavedSteps=numSavedSteps , thinSteps=thinSteps ,   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 10:09:21 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 136  
  
WARNING: Unused variable(s) in data table:  
ySum  
  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 1000  
-------------------------------------------------| 1000  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 2000  
-------------------------------------------------| 2000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . . . . . . . Failed to set trace monitor for delta  
Node not found  
. Updating 40000  
-------------------------------------------------| 40000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . Updating 0  
. Deleting model  
.   
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
summaryInfo = smryMCMC\_cont\_table( mcmcCoda ,   
 datFrm=myDataFrame , x1Name=x1Name , x2Name=x2Name ,  
 x1contrasts=x1contrasts  
 )  
  
# # # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_cont\_table( mcmcCoda ,   
# datFrm=myDataFrame , yName=yName , x1Name=x1Name ,   
# x2Name=x2Name ,  
# x1contrasts=x1contrasts )  
  
# table of retouch artefact counts  
kable(rt\_tab, caption = "Table of frequencies of retouched flakes by phase")

Table of frequencies of retouched flakes by phase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | phase 1 | phase 2 | phase 3 | phase 4 |
| rt.Retype | 32 | 21 | 59 | 57 |
| flakes | 475 | 371 | 1130 | 915 |

# subset summaryInfo to show HDI interval for interactions  
 HDI\_intervals\_for\_interactions\_fr <- summaryInfo\_ANOVA[c((nrow(summaryInfo\_ANOVA)-length(contrasts)+1):nrow(summaryInfo\_ANOVA)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions\_fr,3), caption = "Highest density intervals (95%) for the posterior distributions of the interactions between phases and retouched flake frequency.")

Highest density intervals (95%) for the posterior distributions of the interactions between phases and retouched flake frequency.

|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| 1.v.2 | -2.967 | 9.376 |
| 2.v.3 | -11.287 | 1.707 |
| 3.v.4 | -1.208 | 8.323 |

The results of the code chunk above indicates that there are credible differences in the frequencies of retouched artefacts during phase two to three (altough this interval is very close to zero), and during phase three to four. A look at the raw counts shows that from phase three to four there is a decrease in the proportion of retouched pieces, but that the overall count is still low.

chert\_artefacts\_retouch\_nhst <- assocstats(as.matrix(rt\_tab))

The code chunk above returns a chi-squared value of 1.5586223 and a p-value of 0.6688079, indicating a nonsignficant difference in frequencies of breakage classes by phase. The Cramer's V value of 0.0225689 indicates that the effect size is extremely small. We interpret this result to mean that there is no substantial significance in the differences in frequencies of retouched flakes by phase.

The code chunk below investigates changes in the length of the retouched margin of artefacts over time.

rt\_len <- data.frame(phase = rt$phase, rt\_len = rt$Retlen)  
# omit NAs  
rt\_len <- rt\_len[!(is.na(rt\_len$rt\_len)), ]  
# get mean lengths  
mean\_lengths <- aggregate(rt\_len ~ phase, rt\_len, mean)  
# do bayesian ANOVA  
myDataFrame <- rt\_len  
myDataFrame <- myDataFrame[complete.cases(myDataFrame),] # omit NAs  
yName = names(myDataFrame)[2] # retouch length  
xName = names(myDataFrame)[1] # phase  
contrasts = list(  
 list( c("1") , c("2") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("2") , c("3") , compVal=0.0 , ROPE=c(-0.1,0.1)) ,  
 list( c("3") , c("4") , compVal=0.0 , ROPE=c(-0.1,0.1)) )  
# Generate the MCMC chain:  
 mcmcCoda\_ANOVA = genMCMC\_ANOVA(datFrm=myDataFrame , yName=yName , xName=xName ,  
 numSavedSteps=11000 , thinSteps=10   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 11:58:38 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 214  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 500  
-------------------------------------------------| 500  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 1000  
-------------------------------------------------| 1000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . Updating 36670  
-------------------------------------------------| 36650  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
\* 100%  
. . . . Updating 0  
. Deleting model  
.   
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
 summaryInfo\_ANOVA = smryMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
 datFrm=myDataFrame , xName=xName ,  
 contrasts=contrasts   
 )  
 # # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
# datFrm=myDataFrame , yName=yName , xName=xName ,   
# contrasts=contrasts )  
#   
# subset summaryInfo to show HDI interval for interactions  
 HDI\_intervals\_for\_interactions\_l <- summaryInfo\_ANOVA[c((nrow(summaryInfo\_ANOVA)-length(contrasts)+1):nrow(summaryInfo\_ANOVA)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions\_l ,3), caption = "Highest density intervals (95%) for the posterior distributions of the interactions between phases and retouch length")

Highest density intervals (95%) for the posterior distributions of the interactions between phases and retouch length

|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| 1.v.2 | -3.004 | 6.555 |
| 2.v.3 | -8.623 | 2.138 |
| 3.v.4 | -4.073 | 6.271 |

# combine retouch freq and length into one table  
HDIs\_flake\_retouch\_freq\_and\_length <- round(cbind(HDI\_intervals\_for\_interactions\_fr, HDI\_intervals\_for\_interactions\_l),3)  
kable(HDIs\_flake\_retouch\_freq\_and\_length, caption = "Highest density intervals (95%) for the posterior distributions of the interactions between phases and retouch frequency and length")

Highest density intervals (95%) for the posterior distributions of the interactions between phases and retouch frequency and length

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | HDIlow | HDIhigh | HDIlow | HDIhigh |
| 1.v.2 | -2.967 | 9.376 | -3.004 | 6.555 |
| 2.v.3 | -11.287 | 1.707 | -8.623 | 2.138 |
| 3.v.4 | -1.208 | 8.323 | -4.073 | 6.271 |

The results of the code chunk above show that the HDIs for all interactions include zero. This indicates no credible difference in retouch length by phase.

The code chunk below repeats this analysis using a frequentist test.

# ANOVA with Tukey's HSD  
fit <- aov(rt\_len ~ as.factor(phase), myDataFrame)  
fit\_summary <- summary(fit)  
tuk <- TukeyHSD(fit)  
# plot(tuk, las = 2, cex = 0.1)  
kable(round(tuk$`as.factor(phase)`,3), caption = "Tukey's Honest Significant Difference for phase by phase comparisons of flake retouch length")

Tukey's Honest Significant Difference for phase by phase comparisons of flake retouch length

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | diff | lwr | upr | p adj |
| 2-1 | -2.956 | -11.071 | 5.158 | 0.775 |
| 3-1 | 2.043 | -6.320 | 10.405 | 0.919 |
| 4-1 | 0.210 | -8.293 | 8.713 | 1.000 |
| 3-2 | 4.999 | -4.032 | 14.029 | 0.472 |
| 4-2 | 3.166 | -5.995 | 12.328 | 0.802 |
| 4-3 | -1.833 | -11.214 | 7.549 | 0.956 |

In the code chunk above we obtain the results of a frequentist ANOVA which returns a statistically nonsignificant result: F = 0.7300247, df = 3, p = 0.5369313. The Tukey's Honest Significant Difference test shows confidence intervals on the differences between the means artefact retouch length of a phase. None of the interactions show a significant difference. We interpret this as evidence of overall no change in flake retouch length.

In the code chunk below we compute a Bayesian t-test to investigate differences in the mass of retouched and non-retouched flakes.

# Bayesian t-test of the mass of all retouched vs all unretouched flakes  
# flakes\_retouch\_size\_t\_test <- BESTmcmc(flakes\_retouch\_size[flakes\_retouch\_size$Artclas == "flake", ]$Weight, flakes\_retouch\_size[flakes\_retouch\_size$Artclas == "retf", ]$Weight)

In the code chunk below we perform a frequentist t-test on flake mass of retouched and non-retouched flakes.

retouch\_nonretouch\_flake\_mass\_nhst <- t.test(flakes\_retouch\_size[flakes\_retouch\_size$Artclas == "flake", ]$Weight, flakes\_retouch\_size[flakes\_retouch\_size$Artclas == "retf", ]$Weight)

The code chunk above returns the result of the frequentist t-test as follows: t = -6.1886492, df = 158.102378 and p = 5.000649310^{-9}.

The code chunk below makes a table the summarises retouch indices for retouched pieces recovered from Jerimalai. GIUR = Geometric Index of Unifacial Retouch, II = Index of Invasiveness, % = percent of perimeter with retouch

retouch\_indices[is.na(retouch\_indices)] <- 0  
retouch\_indices$GIUR <- with(retouch\_indices, t1/T1 + t2/T2 + t3/T3)/3  
retouch\_indices$perimeter\_perc <- with(retouch\_indices, length/perimeter \* 100)  
retouch\_indices$II <- with(retouch\_indices, ((X0.5 \* 0.5) + (X1 \* 1))/16)  
# get mean and standard deviation for each index  
retouch\_indices\_subset <- retouch\_indices %>% select(GIUR, perimeter\_perc, II)   
# sweep over the columns to compute mean and standard deviation  
retouch\_indices\_means <- data.frame(t(round(apply(retouch\_indices\_subset, 2, mean, na.rm = TRUE),2)))  
retouch\_indices\_sds <- data.frame(t(round(apply(retouch\_indices\_subset, 2, sd, na.rm = TRUE),2)))  
# make table  
retouch\_table <- retouch\_indices %>%  
 select(Square, Spit, Type, GIUR, II, perimeter\_perc)  
# do some rounding  
retouch\_table[,4:6] <- apply(retouch\_table[,4:6], 2, round, 2)  
# have a look  
retouch\_table <- arrange(retouch\_table, Spit)  
retouch\_table$Phase <- makephases(retouch\_table$Spit)  
retouch\_table <- retouch\_table[, c(1,2,7,3:6)]  
# write the table to a csv file so we can put it in the word doc  
write.csv(retouch\_table, file = 'retouch\_table.csv', row.names = FALSE)  
kable(retouch\_table, caption = "Summary of retouch indices for retouched pieces recovered from Jerimalai. GIUR = Geometric Index of Unifacial Retouch, II = Index of Invasiveness, % = percent of perimeter with retouch")

Summary of retouch indices for retouched pieces recovered from Jerimalai. GIUR = Geometric Index of Unifacial Retouch, II = Index of Invasiveness, % = percent of perimeter with retouch

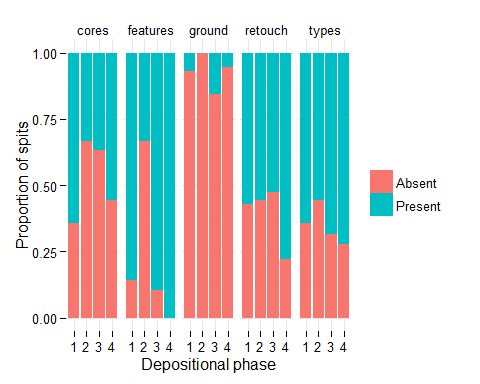
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Square | Spit | Phase | Type | GIUR | II | perimeter\_perc |
| B | 3 | 4 | Ventral Side | NaN | 0.16 | 44.44 |
| B | 3 | 4 | Side and End | 0.42 | 0.16 | 28.70 |
| B | 4 | 4 | Double side and end (steep edged) | 0.83 | 0.28 | 50.00 |
| B | 4 | 4 | Double Side | 0.85 | 0.34 | 33.04 |
| B | 5 | 4 | Side Ventral | NaN | 0.06 | 11.94 |
| B | 10 | 4 | Side | 0.37 | 0.09 | 18.66 |
| B | 10 | 4 | End | 0.46 | 0.03 | 11.80 |
| B | 12 | 4 | Drill? | 0.55 | 0.19 | 43.10 |
| B | 14 | 4 | Notch | 0.62 | 0.09 | 20.00 |
| B | 19 | 4 | Denticulate | 0.19 | 0.16 | 27.68 |
| B | 22 | 3 | Notch (Ventral) | NaN | 0.09 | 40.69 |
| B | 25 | 3 | Bifacial End | 0.37 | 0.09 | 17.36 |
| B | 34 | 3 | Side and End (Alternating bifacial) | 0.38 | 0.12 | 28.23 |
| B | 38 | 3 | Notch (Ventral) | NaN | 0.06 | 14.59 |
| B | 39 | 3 | Concave Bifacial side and end | 0.80 | 0.56 | 60.15 |
| B | 40 | 2 | Side | 0.26 | 0.09 | 21.31 |
| A | 43 | 2 | Bec | 0.52 | 0.16 | 29.51 |

The retouch intensity can be summarised with the following metrics: GIUR = 0.51 +/- 0.21 perimeter = 29.48 +/- 14.09% II = 0.16 +/- 0.13

# Results: Technological types

This code makes the table "Summary of counts and classes. Counts refers to the count of spits in each depositional phase containing a given class."

# all   
  
# combine  
  
techno <- data.frame(cores, types, retouch, features, ground, stringsAsFactors = FALSE)  
# remove extra Spit cols (is those called 'Spit.1' etc)  
techno <- techno[,-grep("Spit\\.", colnames(techno)) ]  
# put depths on   
techno$depth <- depths$Depth.bs..m[match(techno$Spit,depths$Spit.no)]  
# put phases on   
techno$phase <- flakes$phase[match(techno$Spit,flakes$Spit)]  
  
techno[] <- lapply(techno, as.character) # change factor to character  
techno[techno == 'x'] <- 1 # replace x with 1  
techno[] <- lapply(techno, as.numeric) # change char to num  
# get row sums  
rs <- rowSums(techno[,2:32], na.rm = TRUE )  
# get row sums by group  
rs\_phase <- data.frame(rs, phase = techno$phase)  
# % that have a type present  
# phases with no types at all  
check <- rs\_phase[rs\_phase$rs == 0,]  
# set 1 or above to 1  
rs\_phase$rs <- ifelse(rs\_phase$rs == 0, 0, 1)  
check <- aggregate(rs ~ phase, rs\_phase[rs\_phase$rs == 0,], length) # counts  
check <- t(apply(table(rs\_phase), 2, function(x) x/sum(x))) # props  
# just three spits with zero types... let's do it by class  
  
## cores ##  
# put depths on   
cores$depth <- depths$Depth.bs..m[match(cores$Spit,depths$Spit.no)]  
# put groups on   
cores$phase <- flakes$phase[match(cores$Spit,flakes$Spit)]  
cores[] <- lapply(cores, as.character) # change factor to character  
cores[cores == 'x'] <- 1 # replace x with 1  
cores[] <- lapply(cores, as.numeric) # change char to num  
# get row sums  
rs <- rowSums(cores[,2:(ncol(cores)-2)], na.rm = TRUE )  
# get row sums by phase  
rs\_phase <- data.frame(rs, group = cores$phase)  
# % that have a type present  
# groups with no types at all  
check <- rs\_phase[rs\_phase$rs == 0,]  
# set any non-zero to 1  
rs\_phase$rs <- ifelse(rs\_phase$rs != 0, 1, rs\_phase$rs)  
# yes, more with zero here...  
dc\_core <- data.frame(t(apply(table(rs\_phase), 2, function(x) x/sum(x))))  
dc\_core <- prop.table(as.matrix(dc\_core), 1)  
  
## types ##  
# put depths on   
types$depth <- depths$Depth.bs..m[match(types$Spit,depths$Spit.no)]  
# put groups on   
types$phase <- flakes$phase[match(types$Spit,flakes$Spit)]  
types[] <- lapply(types, as.character) # change factor to character  
types[types == 'x'] <- 1 # replace x with 1  
types[] <- lapply(types, as.numeric) # change char to num  
# get row sums  
rs <- rowSums(types[,2:(ncol(types)-2)], na.rm = TRUE )  
# get row sums by group  
rs\_phase <- data.frame(rs, phase = types$phase)  
# % that have a type present  
# groups with no types at all  
check <- rs\_phase[rs\_phase$rs == 0,]  
# set any non-zero to 1  
rs\_phase$rs <- ifelse(rs\_phase$rs != 0, 1, rs\_phase$rs)  
# yes, more with zero here...  
dc\_types <- data.frame(t(apply(table(rs\_phase), 2, function(x) x/sum(x))))  
dc\_types <- prop.table(as.matrix(dc\_types), 1)  
  
## retouch ##  
# put depths on   
retouch$depth <- depths$Depth.bs..m[match(retouch$Spit,depths$Spit.no)]  
# put groups on   
retouch$phase <- flakes$phase[match(retouch$Spit,flakes$Spit)]  
retouch[] <- lapply(retouch, as.character) # change factor to character  
retouch[retouch == 'x'] <- 1 # replace x with 1  
retouch[] <- lapply(retouch, as.numeric) # change char to num  
# get row sums  
rs <- rowSums(retouch[,2:(ncol(retouch)-2)], na.rm = TRUE )  
# get row sums by group  
rs\_phase <- data.frame(rs, phase = retouch$phase)  
# % that have a type present  
# groups with no types at all  
check <- rs\_phase[rs\_phase$rs == 0,]  
# set any non-zero to 1  
rs\_phase$rs <- ifelse(rs\_phase$rs != 0, 1, rs\_phase$rs)  
# yes, more with zero here...  
dc\_retouch <- data.frame(t(apply(table(rs\_phase), 2, function(x) x/sum(x))))  
dc\_retouch <- prop.table(as.matrix(dc\_retouch), 1)  
  
## features ##  
# put depths on   
features$depth <- depths$Depth.bs..m[match(features$Spit,depths$Spit.no)]  
# put groups on   
features$phase <- flakes$phase[match(features$Spit,flakes$Spit)]  
features[] <- lapply(features, as.character) # change factor to character  
features[features == 'x'] <- 1 # replace x with 1  
features[] <- lapply(features, as.numeric) # change char to num  
# get row sums  
rs <- rowSums(features[,2:(ncol(features)-2)], na.rm = TRUE )  
# get row sums by group  
rs\_phase <- data.frame(rs, phase = features$phase)  
# % that have a type present  
# groups with no types at all  
check <- rs\_phase[rs\_phase$rs == 0,]  
# set any non-zero to 1  
rs\_phase$rs <- ifelse(rs\_phase$rs != 0, 1, rs\_phase$rs)  
# yes, more with zero here...  
dc\_feat <- data.frame(t(apply(table(rs\_phase), 2, function(x) x/sum(x))))  
dc\_feat <- prop.table(as.matrix(dc\_feat), 1)  
  
## ground ##  
# put depths on   
ground$depth <- depths$Depth.bs..m[match(ground$Spit,depths$Spit.no)]  
# put phases on   
ground$phase <- flakes$phase[match(ground$Spit,flakes$Spit)]  
ground[] <- lapply(ground, as.character) # change factor to character  
ground[ground == 'x'] <- 1 # replace x with 1  
ground[] <- lapply(ground, as.numeric) # change char to num  
# get row sums  
rs <- rowSums(ground[,2:(ncol(ground)-2)], na.rm = TRUE )  
# get row sums by phase  
rs\_phase <- data.frame(rs, phase = ground$phase)  
# % that have a type present  
# phases with no types at all  
check <- rs\_phase[rs\_phase$rs == 0,]  
# set any non-zero to 1  
rs\_phase$rs <- ifelse(rs\_phase$rs != 0, 1, rs\_phase$rs)  
# yes, more with zero here...  
dc\_gr <- data.frame(t(apply(table(rs\_phase), 2, function(x) x/sum(x))))  
dc\_gr <- prop.table(as.matrix(dc\_gr), 1)  
  
# put them together  
lst <- list(cores = dc\_core, retouch = dc\_retouch, types = dc\_types, features = dc\_feat, ground = dc\_gr)  
df <- ldply(lst, data.frame)  
df$phase <- rep(seq\_along(unique(na.omit(flakes$phase))), length(lst))  
  
# plot proportion of spits having a techno-type present  
df\_m <- melt(df, id.var = c('phase', '.id'))  
df\_m$variable <- ifelse(df\_m$variable == 'X0', 'Absent', 'Present')  
ggplot(df\_m, aes(as.factor(phase), value, fill = variable)) +  
 geom\_bar(stat="identity") +  
 facet\_grid(. ~ .id) +  
 theme\_minimal() +  
 xlab("Depositional phase") +  
 ylab("Proportion of spits") +  
 scale\_fill\_discrete(name="")



# now we can see, let's explore some of the minor patterns...  
# what are the counts of each class in each phase?  
  
# more about features  
l <- lapply(features[,2:(ncol(features)-2)], function(i) aggregate( i ~ phase, features, sum, na.rm = TRUE))  
df <- do.call(rbind.data.frame, l)  
df$name <- unlist(lapply(1:length(l), function(i) rep(names(l)[i], nrow(l[[i]]))))  
feat <- dcast(phase ~ name, value.var = 'i', data = df)  
  
# more about ground  
l <- lapply(ground[,2:(ncol(ground)-2)], function(i) aggregate( i ~ phase, ground, sum, na.rm = TRUE))  
df <- do.call(rbind.data.frame, l)  
df$name <- unlist(lapply(1:length(l), function(i) rep(names(l)[i], nrow(l[[i]]))))  
grou <- dcast(phase ~ name, value.var = 'i', data = df)  
  
# more about retouch  
l <- lapply(retouch[,2:(ncol(retouch)-2)], function(i) aggregate( i ~ phase, retouch, sum, na.rm = TRUE))  
df <- do.call(rbind.data.frame, l)  
df$name <- unlist(lapply(1:length(l), function(i) rep(names(l)[i], nrow(l[[i]]))))  
reto <- dcast(phase ~ name, value.var = 'i', data = df)  
  
# more about types  
l <- lapply(types[,2:(ncol(types)-2)], function(i) aggregate( i ~ phase, types, sum, na.rm = TRUE))  
df <- do.call(rbind.data.frame, l)  
df$name <- unlist(lapply(1:length(l), function(i) rep(names(l)[i], nrow(l[[i]]))))  
type <- dcast(phase ~ name, value.var = 'i', data = df)  
  
# more about cores  
l <- lapply(cores[,2:(ncol(cores)-2)], function(i) aggregate( i ~ phase, cores, sum, na.rm = TRUE))  
df <- do.call(rbind.data.frame, l)  
df$name <- unlist(lapply(1:length(l), function(i) rep(names(l)[i], nrow(l[[i]]))))  
core <- dcast(phase ~ name, value.var = 'i', data = df)  
  
# obsidian?  
all$phase <- makephases(all$Spit)  
obsidian <- table(data.frame(phase = all$phase, rm = all$Material))  
  
# summary table of all techno-types. This will give a count of spits  
# in each phase that contain at least one artefact in the category.  
summaryt <- (t(cbind(  
 ddply(retouch[,2:(ncol(retouch))], "phase", numcolwise(sum, na.rm = TRUE)),  
 ddply(features[,2:(ncol(features))], "phase", numcolwise(sum, na.rm = TRUE)),  
 ddply(ground[,2:(ncol(ground))], "phase", numcolwise(sum, na.rm = TRUE)),  
 ddply(types[,2:(ncol(types))], "phase", numcolwise(sum, na.rm = TRUE)),  
 ddply(cores[,2:(ncol(cores))], "phase", numcolwise(sum, na.rm = TRUE))  
)))  
summaryt <- summaryt[!(row.names(summaryt) %in% c("depth", "phase")),]  
# compute proportions so we have the proportion of spits in each phase   
# that contains at least one of each class.   
spits\_per\_phase <- aggregate(Spit ~ phase, data = cores, length)  
summaryt\_props <- data.frame(round(t(t(summaryt) / spits\_per\_phase$Spit),2))  
names(summaryt\_props) <- c( "phase 1", "phase 2", "phase 3", "phase 4")  
# write table to csv to put into word doc  
write.csv(summaryt\_props, "techno\_types\_table.csv")  
kable(summaryt\_props, caption = "Summary of proportions and classes. Proportions refers to the proportion of spits in each depositional phase containing a given class")

Summary of proportions and classes. Proportions refers to the proportion of spits in each depositional phase containing a given class

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | phase 1 | phase 2 | phase 3 | phase 4 | NA |
| Distal | 0.36 | 0.11 | 0.11 | 0.56 | 0.00 |
| Lateral | 0.22 | 0.11 | 0.22 | 0.57 | 0.00 |
| Notched | 0.00 | 0.00 | 0.21 | 0.67 | 0.00 |
| Lateral.and.Distal | 0.11 | 0.07 | 0.00 | 0.05 | 0.00 |
| Double.Side.and.End | 0.14 | 0.11 | 0.05 | 0.22 | 0.00 |
| Denticulated | 0.00 | 0.00 | 0.06 | 0.07 | 0.00 |
| Notched.Lateral.and.Distal | 0.00 | 0.06 | 0.00 | 0.00 | 0.00 |
| Drill.Like.Retouch...Edge.Damage | 0.00 | 0.00 | 0.00 | 0.11 | 0.00 |
| Old.Ventral.Suface.on.Dorsal...Faceted.Platform | 0.64 | 0.11 | 0.53 | 0.39 | 0.07 |
| Platform.Faceting | 1.33 | 0.16 | 0.89 | 1.21 | 0.22 |
| Flake.With.Old.Ventral.on.Dorsal | 0.00 | 0.00 | 0.14 | 0.89 | 0.00 |
| Truncated.Flake | 0.00 | 0.00 | 0.11 | 0.00 | 0.00 |
| Flakes.With.Gloss | 0.00 | 0.00 | 0.05 | 0.17 | 0.00 |
| Possible.Ground.Flake..Axe.Flake.. | 0.11 | 0.00 | 0.00 | 0.00 | 0.00 |
| Striated.haematite | 0.00 | 0.00 | 0.21 | 0.11 | 0.00 |
| Microblade | 0.22 | 0.07 | 0.78 | 0.11 | 0.00 |
| Levallois.Like.Flake | 0.14 | 0.00 | 0.11 | 0.00 | 0.00 |
| Burin.Spall | 0.22 | 0.05 | 0.00 | 0.29 | 0.00 |
| Faceted.Point | 0.16 | 0.00 | 0.00 | 0.00 | 0.00 |
| Bipolar.Flake | 0.00 | 0.07 | 0.22 | 0.21 | 0.00 |
| Eclat.Debordant | 0.00 | 0.11 | 0.21 | 0.00 | 0.00 |
| Cortical.Flake | 0.22 | 0.00 | 0.06 | 0.21 | 0.00 |
| Redirecting.Flake | 0.05 | 0.06 | 0.36 | 1.00 | 0.00 |
| Chopper.Anvil | 0.00 | 0.07 | 0.00 | 0.05 | 0.00 |
| Semi.Discoidal | 0.07 | 0.11 | 0.00 | 0.11 | 0.00 |
| Faceted.Radial...Levallois | 0.44 | 0.00 | 0.06 | 0.21 | 0.00 |
| Truncated.Faceted.Flake | 0.21 | 0.00 | 0.07 | 0.11 | 0.00 |
| Bipolar | 0.11 | 0.00 | 0.11 | 0.21 | 0.00 |
| Bidirectional | 0.07 | 0.11 | 0.05 | 0.11 | 0.00 |
| Single.Platform | 0.22 | 0.00 | 0.06 | 0.29 | 0.00 |
| Mulitplatform | 0.21 | 0.06 | 0.21 | 0.44 | 0.00 |

# Summary of results for Stone Artefacts from Jeremalai Square A

sq\_a\_all <- read.csv("data/Jerimalai\_All\_Artefacts\_Square\_A.csv") %>%   
 filter(Square == "A")

## Chronology

Radiocarbon ages from square A have fewer Pleistocene dates than square B and an inversion of samples from spits 38 and 27. These details make it difficult to group excavated spits into analytical units for comparison with square B. The stone artefact assemblage is also smaller than from square B. Due to these limitations, we took a conservative approach and aggreagates spits from square A into two groups: a Pleistocene deposit that includes spits 61-26 and a Holocene deposit that includes spits 25-1.

sq\_a\_radiocarbon <- read.csv("data/Jerimalai\_dates\_Square\_A.csv")  
kable(sq\_a\_radiocarbon, caption = "Radiocarbon ages from Jerimalai square A")

Radiocarbon ages from Jerimalai square A

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| spit | material | age | error | cal\_age\_min | cal\_age\_max | lab\_code | depth\_bs |
| 3 | Charcoal | 2570 | 34 | 2695 | 2759 | Wk-19224 | NA |
| 5 | Charcoal | 3245 | 39 | 3385 | 3562 | Wk-19225 | NA |
| 6 | Turbo | 5341 | 41 | 5598 | 5835 | Wk-18154 | NA |
| 12 | Turbo | 5567 | 44 | 5866 | 6101 | Wk-18155 | NA |
| 13 | Turbo | 5549 | 62 | 5762 | 6115 | Wk-17829 | NA |
| 21 | Turbo | 5909 | 40 | 6245 | 6422 | Wk-19226 | NA |
| 26 | Haliotis cf. varia Turbo | 10110 | 79 | 10870 | 11245 | Wk-18156 | NA |
| 27 | Turbo | 19952 | 235 | 22564 | 23935 | Wk-17830 | NA |
| 38 | Turbo | 13658 | 91 | 15455 | 16761 | Wk-19227 | NA |
| 46 | Trochus | 38255 | 596 | 41616 | 43381 | Wk-17831 | NA |

## Results: Raw materials

The output from the code chunk below shows that chert strongly dominates raw material choices in square A, just as it did in square B.

# subset flakes  
sq\_a\_flakes <- sq\_a\_all %>%   
 filter(Artclas == "Flake")  
  
  
# In Square A, Phase I corresponds to Spits 46-39 and has an associated date of 42 ka from Spit 39 (Table 1). Phase II is comprised of Spits 38-26 with associated ages of between 17-10 ka (with one inversion noted in Spit 27 with a date of 22 ka). Phase III consists of Spits 25-6 and is associated with ages of 6.4-5.5 ka, and Phase IV is covered by Spits 5-1 with associated ages of 3-0 ka.   
  
# here's a function to assign phases based on spit numbers  
 sq\_a\_makephases <- function(x) {ifelse(x >= 1 & x <= 5, 4,  
 ifelse(x >= 6 & x <= 25, 3,  
 ifelse(x >= 26 & x <= 38, 2,  
 ifelse(x >= 39 & x <= 46, 1, NA))))}  
# apply phase numbers  
sq\_a\_flakes <- sq\_a\_flakes[!(is.na(sq\_a\_flakes$Spit)),]  
sq\_a\_flakes$phase <- sq\_a\_makephases(sq\_a\_flakes$Spit)  
# raw material  
raw <- dcast(sq\_a\_flakes, Material ~ phase)   
# subset dominant raw materials  
raw[,2:3] <- sapply(raw[,2:3], as.numeric)  
dom <- raw[rowSums(raw[,2:3]) > 10,]  
colnames(dom) <- c("raw material", "phase 1", "phase 2", "phase 3", "phase 4")  
kable(dom, caption = "Square A: Frequencies of dominant raw materials by depositional phase")

Square A: Frequencies of dominant raw materials by depositional phase

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | raw material | phase 1 | phase 2 | phase 3 | phase 4 |
| 1 | Chert | 277 | 774 | 798 | 120 |
| 10 | Volcanic | 6 | 8 | 28 | 21 |

data <- melt(as.matrix(dom), varnames=c("raw\_material", "phase"), value.name="Freq")  
myDataFrame = data  
yName=names(myDataFrame)[3] # Freq  
x1Name=names(myDataFrame)[2] # phase   
x2Name=names(myDataFrame)[1] # raw material  
x1contrasts = list(   
 list( c("phase 1") , c("phase 2") , compVal=0.0 , ROPE=c(-0.1,0.1) ),  
 list( c("phase 2") , c("phase 3") , compVal=0.0 , ROPE=c(-0.1,0.1) ),  
 list( c("phase 3") , c("phase 4") , compVal=0.0 , ROPE=c(-0.1,0.1) ))  
  
numSavedSteps = 12000 # MCMC parameters  
thinSteps = 10 # MCMC parameters  
  
mcmcCoda = genMCMC\_cont\_table( datFrm=myDataFrame ,   
 yName=yName , x1Name=x1Name , x2Name=x2Name ,  
 numSavedSteps=numSavedSteps , thinSteps=thinSteps ,   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 10:09:42 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 160  
  
WARNING: Unused variable(s) in data table:  
ySum  
  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 1000  
-------------------------------------------------| 1000  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 2000  
-------------------------------------------------| 2000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . . . . . . . Failed to set trace monitor for delta  
Node not found  
. Updating 40000  
-------------------------------------------------| 40000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . Updating 0  
. Deleting model  
.   
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
summaryInfo = smryMCMC\_cont\_table( mcmcCoda ,   
 datFrm=myDataFrame , x1Name=x1Name , x2Name=x2Name ,  
 x1contrasts=x1contrasts   
 )  
  
# # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_cont\_table( mcmcCoda ,   
# datFrm=myDataFrame , yName=yName , x1Name=x1Name ,   
# x2Name=x2Name ,  
# x1contrasts=x1contrasts )  
  
# subset summaryInfo to show HDI interval for interactions  
HDI\_intervals\_for\_interactions <- summaryInfo[c((nrow(summaryInfo) - length(x1contrasts) + 1):nrow(summaryInfo)), c("HDIlow", "HDIhigh")]  
  
kable(round(HDI\_intervals\_for\_interactions,3), caption = "Square A: Highest density intervals (95%) for the posterior distributions of the interactions between phases and raw material frequencies.")

Square A: Highest density intervals (95%) for the posterior distributions of the interactions between phases and raw material frequencies.

|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| phase 1.v.phase 2 | -1.128 | 0.714 |
| phase 2.v.phase 3 | -1.128 | 0.508 |
| phase 3.v.phase 4 | -0.503 | 1.183 |

# here is the frequentist equivalent   
sq\_A\_raw\_material\_by\_phase\_nhst <- assocstats(as.matrix(dom[,-1]))

The Bayesian test includes zero in the HDI, indicating a non-credible difference. The t-test In the code chunk above we obtain the results of a frequentist t-test which returns a statistically significant result: t =

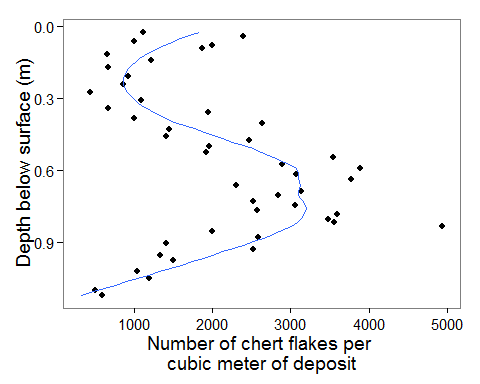
|  |  |  |  |
| --- | --- | --- | --- |
| 277 6 ---- | 774 8 ---- | 798 28 ---- | 120 21 ----, df = |

51.10428 3 0 77.64331 3 0 --------- --- ---, p = NA. This shows a statistically significant change in flake raw material, however the low Cramer's V indicates a very small effect size. This is consistant with what we see in the Bayesian HDI intervals, and we interpret these results as an insubstantial increase in the very small proportion of artefacts made from volcanic rock.

## Results: Discard rates

The code chunk below generates the figure that shows discard rates of chert artefacts over time at Jerimalai square A. Each point is an excavation unit. The blue line is a locally weighted regression line (span = 0.4) to aid in visualising the trend. Artefact discard at square A peaksat aroudn 0.7m below the surface, an equivalent depth to the peak in square B. This corresponds to around spits 33-34, which lies between the two inverted dates (Wk-17830, Wk-19227). This means that we can only offer a wide interval of 20-15 ka BP for the age of the peak discard at square A. This is broadly consistent with a post-LGM increase in discard observed at square B.

# spit depths  
sq\_a\_depths <- read.csv("data/Jerimalai\_spit\_depths\_Square\_A.csv")  
# sediment volumes  
vols <- read.csv("data/Artefact densities with soil volumes Sq B.csv", skip = 1)  
# put depths on lithic data  
sq\_a\_flakes$depth <- sq\_a\_depths$Depth.bs..m[match(sq\_a\_flakes$Spit,sq\_a\_depths$Spit.no)]  
# discard rates  
sq\_a\_discard <- aggregate(Weight ~ depth + Spit, sq\_a\_flakes, length)  
# sediment volumes: put volumes on  
sq\_a\_discard$sedvol <- vols$Soil.1[match(sq\_a\_discard$Spit, vols$X.3)]  
# put spit thickesses on  
sq\_a\_discard$thick <- c(0.018, diff(sq\_a\_discard$depth)) # add first value from depth\_and\_dates.xsl  
# compute artefacts per kg of sediment  
sq\_a\_discard$kgsed <- with(sq\_a\_discard, Weight / sedvol) # weight is count of artefacts that have a weight  
# compute artefact per cubic meter (spit thickess)  
sq\_a\_discard$cubmet <- with(sq\_a\_discard, Weight / thick)  
# seems we have an unusually extreme value in spit 34  
# omit - perhaps a data collection typo  
# discard <- discard[discard$Spit != 34, ]  
# Plot  
ggplot(sq\_a\_discard, (aes(depth, cubmet))) +  
 geom\_point() +  
 stat\_smooth(span = 0.5, se = FALSE) +  
 xlab("Depth below surface (m)") +  
 ylab("Number of chert flakes per \ncubic meter of deposit") +  
 coord\_flip() +  
 scale\_x\_reverse()



## Results: Taphonomy

The code chunk below computes a Bayesian Poisson exponential ANOVA to investigate differences in flake breakage classes over time.

sq\_a\_allchert <- sq\_a\_all[sq\_a\_all$Material == 'Chert', ]  
sq\_a\_allchert$phase <- sq\_a\_makephases(sq\_a\_allchert$Spit)  
# make Artclass that is long and transv breaks  
sq\_a\_allchert$Artclas <- ifelse(sq\_a\_allchert$Breaks == "",   
 as.character(sq\_a\_allchert$Artclas),   
 paste(sq\_a\_allchert$Artclas, sq\_a\_allchert$Breaks, sep = "-"))  
sq\_a\_taph <- data.frame(table(sq\_a\_allchert$Artclas))  
# use regex to get broken flakes -b-   
sq\_a\_broken <- sq\_a\_allchert[grep("-b", sq\_a\_allchert$Artclas), ]  
# get counts of broken to complete per phase  
# flake to -b-  
sq\_a\_breaks <- dcast(sq\_a\_allchert, Artclas ~ phase)[-1,]  
sq\_a\_breaks <- sq\_a\_breaks[sq\_a\_breaks$Artclas =="flake" | grepl("-b-", sq\_a\_breaks$Artclas), ]  
sq\_a\_allchert$Artclas <- tolower(sq\_a\_allchert$Artclas)  
sq\_a\_allchert$breakt <- "" # create variable to fill  
sq\_a\_allchert$breakt[grep("trans", sq\_a\_allchert$Artclas)] <- "trans"  
sq\_a\_allchert$breakt[grep("long", sq\_a\_allchert$Artclas)] <- "long"  
# per depositional phase  
sq\_a\_breakt <- dcast(sq\_a\_allchert, breakt ~ phase)[-1,]  
# add complete flake counts  
sq\_a\_breakt <- rbind( sq\_a\_breakt , setNames( sq\_a\_breaks[1, ] , names( sq\_a\_breakt ) ) )  
# shift rownames out and delete them  
rownames(sq\_a\_breakt) <- sq\_a\_breakt[,1]  
sq\_a\_breakt <- sq\_a\_breakt[,-1]  
# do bayesian contingency table test  
colnames(sq\_a\_breakt) <- c("phase 1", "phase 2", "phase 3", "phase 4")  
data <- melt(as.matrix(sq\_a\_breakt), varnames=c("breakt", "phase"), value.name="Freq")  
myDataFrame = data  
yName=names(myDataFrame)[3] # Freq   
x1Name=names(myDataFrame)[2] # phase  
x2Name=names(myDataFrame)[1] # break type  
x1contrasts = list(   
 list( c("phase 1") , c("phase 2") , compVal=0.0 , ROPE=c(-0.1,0.1)),  
 list( c("phase 2") , c("phase 3") , compVal=0.0 , ROPE=c(-0.1,0.1)),  
 list( c("phase 3") , c("phase 4") , compVal=0.0 , ROPE=c(-0.1,0.1))  
 )  
numSavedSteps = 12000 # MCMC parameters  
thinSteps = 10 # MCMC parameters  
  
mcmcCoda = genMCMC\_cont\_table( datFrm=myDataFrame ,   
 yName=yName , x1Name=x1Name , x2Name=x2Name ,  
 numSavedSteps=numSavedSteps , thinSteps=thinSteps ,   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 10:09:55 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 178  
  
WARNING: Unused variable(s) in data table:  
ySum  
  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 1000  
-------------------------------------------------| 1000  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 2000  
-------------------------------------------------| 2000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . . . . . . . Failed to set trace monitor for delta  
Node not found  
. Updating 40000  
-------------------------------------------------| 40000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . Updating 0  
. Deleting model  
.   
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
summaryInfo = smryMCMC\_cont\_table( mcmcCoda ,   
 datFrm=myDataFrame , x1Name=x1Name , x2Name=x2Name ,  
 x1contrasts=x1contrasts   
 )  
  
# # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_cont\_table( mcmcCoda ,   
# datFrm=myDataFrame , yName=yName , x1Name=x1Name ,   
# x2Name=x2Name ,  
# x1contrasts=x1contrasts )  
  
# table of raw counts  
kable(sq\_a\_breakt, caption = "Square A: Table of frequencies of each class of breakage by phase")

Square A: Table of frequencies of each class of breakage by phase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | phase 1 | phase 2 | phase 3 | phase 4 |
| long | 27 | 35 | 62 | 8 |
| trans | 67 | 179 | 195 | 20 |
| Flake-b-Both | 21 | 39 | 41 | 4 |

# subset summaryInfo to show HDI interval for interactions  
HDI\_intervals\_for\_interactions <- summaryInfo[c((nrow(summaryInfo) - length(x1contrasts) + 1):nrow(summaryInfo)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions,3), caption = "Square A: Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake breakage classes.")

Square A: Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake breakage classes.

|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| phase 1.v.phase 2 | -0.941 | -0.429 |
| phase 2.v.phase 3 | -0.426 | -0.022 |
| phase 3.v.phase 4 | 1.806 | 2.616 |

The above code chunk returns results that there is no credible difference in the frequences of flake breakage types by phase.

The code chunk below computes the frequentist equivalent, the chi-square test and the Cramer's V for effect size.

artefact\_taphonomy\_nhst <- assocstats(as.matrix(sq\_a\_breakt))

The code chunk below summarises the frequencies of heat-treated flakes at Jerimalai square A

check <- sum(sq\_a\_flakes$Heat, na.rm = TRUE) / nrow(sq\_a\_flakes)  
heat\_sqa <- aggregate(Heat ~ phase, sq\_a\_flakes, length)  
total <- aggregate(Spit ~ phase, sq\_a\_flakes, length)  
heat\_sqa$Not\_heat <- total$Spit - heat\_sqa$Heat  
# show proportions that are heat-treated  
max\_heat <- max(heat\_sqa$Heat / total$Spit)  
min\_heat <- min(heat\_sqa$Heat / total$Spit)  
heat\_t <- t(heat\_sqa)  
# do bayesian contingency table test  
names(heat\_t) <- c("phase 1", "phase 2", "phase 3", "phase 4")  
heat <- heat\_t[-1,]

The code chunk below computes a Bayesian Poisson exponential ANOVA to investigate differences in heat treatment by phase.

data <- melt(as.matrix(heat), varnames=c("heat", "phase"), value.name="Freq")  
myDataFrame = data  
yName=names(myDataFrame)[3] # Freq   
x1Name=names(myDataFrame)[2] # phase  
x2Name=names(myDataFrame)[1] # heat  
x1contrasts = list(   
 list( c("phase 1") , c("phase 2") , compVal=0.0 , ROPE=c(-0.1,0.1) ),   
 list( c("phase 2") , c("phase 3") , compVal=0.0 , ROPE=c(-0.1,0.1) ),   
 list( c("phase 3") , c("phase 4") , compVal=0.0 , ROPE=c(-0.1,0.1) )  
 )  
numSavedSteps = 12000 # MCMC parameters  
thinSteps = 10 # MCMC parameters  
  
mcmcCoda = genMCMC\_cont\_table( datFrm=myDataFrame ,   
 yName=yName , x1Name=x1Name , x2Name=x2Name ,  
 numSavedSteps=numSavedSteps , thinSteps=thinSteps ,   
 )  
  
# Get summary statistics of chain:  
summaryInfo = smryMCMC\_cont\_table( mcmcCoda ,   
 datFrm=myDataFrame , x1Name=x1Name , x2Name=x2Name ,  
 x1contrasts=x1contrasts   
 )  
  
# # # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_cont\_table( mcmcCoda ,   
# datFrm=myDataFrame , yName=yName , x1Name=x1Name ,   
# x2Name=x2Name ,  
# x1contrasts=x1contrasts )  
  
# table of heated artefact counts  
kable(heat, caption = "Square A: Table of frequencies of heat treatment by phase")  
  
# subset summaryInfo to show HDI interval for interactions  
HDI\_intervals\_for\_interactions <- summaryInfo[c((nrow(summaryInfo) - length(x1contrasts) + 1):nrow(summaryInfo)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions,3), caption = "Square A: Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake breakage classes.")

The code chunk above returned results that indicate credible differences in the frequency of heat treatment between phase one and two. There are more flakes that have not been affected by heat in the Holocene phase. The code chunk below computes a frequentist chi-square and Cramer's V test for the heat treatment data.

sq\_A\_chert\_artefacts\_heat\_nhst <- assocstats(as.matrix(heat))

The code chunk above returns a chi-squared value of 18.4834153 and a p-value of 3.49575310^{-4}, indicating a signficant difference in frequencies of breakage classes by phase. However, the Cramer's V value of 0.0947234 indicates that the effect size is extremely small. We interpret this result to mean that although the test result is statistically significant, there is no substantial significance in the differences in frequencies of heat alteration by phase.

## Results: Metric and technological characteristics of cores and unretouched flakes

The code chunk below produces the table summarizing the attributes of chert complete flakes from Jerimalai square A.

metrics <- sq\_a\_flakes %>%   
 group\_by(phase) %>%   
 summarise(mean(Length, na.rm = TRUE),   
 mean(Width, na.rm = TRUE),   
 mean(Thick, na.rm = TRUE),  
 mean(Weight, na.rm = TRUE),  
 mean(Length, na.rm = TRUE),  
 mean(Platwid, na.rm = TRUE),  
 mean(Platthic, na.rm = TRUE),  
 mean(NoDS, na.rm = TRUE),  
 mean(Cortex, na.rm = TRUE),  
 sd(Length, na.rm = TRUE),   
 sd(Width, na.rm = TRUE),   
 sd(Thick, na.rm = TRUE),  
 sd(Weight, na.rm = TRUE),  
 sd(Length, na.rm = TRUE),  
 sd(Platwid, na.rm = TRUE),  
 sd(Platthic, na.rm = TRUE),  
 sd(NoDS, na.rm = TRUE),  
 sd(Cortex, na.rm = TRUE),  
 n = length(Weight))  
  
# get overhang removal data also  
ohr <- filter(sq\_a\_flakes, Overhang == "Yes") %>%  
 group\_by(phase) %>%  
 summarise(OHR\_n = length(Overhang))  
# get percentages of OHR per phase  
ohr$OHR\_perc <- ohr$OHR\_n/metrics$n \* 100  
  
# combine  
metrics <- cbind(metrics, ohr[,c("OHR\_n", "OHR\_perc")])  
metrics <- as.data.frame(t(round(metrics,2)))[-1,]  
colnames(metrics) <- c("phase 1", "phase 2", "phase 3", "phase 4")  
kable(metrics, caption = "Square A: Summary of chert flake metrics by phase")

The code chunk below uses a Bayesian one-way ANOVA to investigate differences in chert flake mass by depositional phase.

myDataFrame <- data.frame(phase = sq\_a\_flakes$phase, mass = sq\_a\_flakes$Weight)  
yName = names(myDataFrame)[2] # mass  
xName = names(myDataFrame)[1] # phase  
contrasts = list(  
 list( c("1") , c("2") , compVal=0.0 , ROPE=c(-0.1,0.1) ),   
 list( c("2") , c("3") , compVal=0.0 , ROPE=c(-0.1,0.1) ),   
 list( c("3") , c("4") , compVal=0.0 , ROPE=c(-0.1,0.1) ))  
# Generate the MCMC chain:  
 mcmcCoda\_ANOVA = genMCMC\_ANOVA(datFrm=myDataFrame , yName=yName , xName=xName ,  
 numSavedSteps=11000 , thinSteps=10   
 )  
 # Get summary statistics of chain:  
 summaryInfo\_ANOVA = smryMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
 datFrm=myDataFrame , xName=xName ,  
 contrasts=contrasts   
 )  
   
# Display posterior information (not easy to read unless using interactively):  
 plotMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
 datFrm=myDataFrame , yName=yName , xName=xName ,   
 contrasts=contrasts )  
   
# subset summaryInfo to show HDI interval for interactions  
HDI\_intervals\_for\_interactions <- summaryInfo\_ANOVA[c((nrow(summaryInfo\_ANOVA) - length(contrasts) + 1):nrow(summaryInfo\_ANOVA)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions,3), caption = "Square A: Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake mass.")

The code chunk above produced results a credibly different distributions of flake mass between the two phases. In the code chunk below we repeat the same investigation using a frequentist ANOVA.

fit <- aov(Weight ~ as.factor(phase), sq\_a\_flakes)  
fit\_summary <- summary(fit)

In the code chunk above we obtain the results of a frequentist t-test which returns a statistically significant result: F = 11.4274362, df = 3, p = 1.97360810^{-7}. This shows a statistically significant change in chert flake mass, consistant with what we see in the Bayesian HDI intervals. Flakes get slightly larger in the Holocene period.

The code chunk below produces a summary table of metric attributes of chert complete cores from Jerimalai square A.

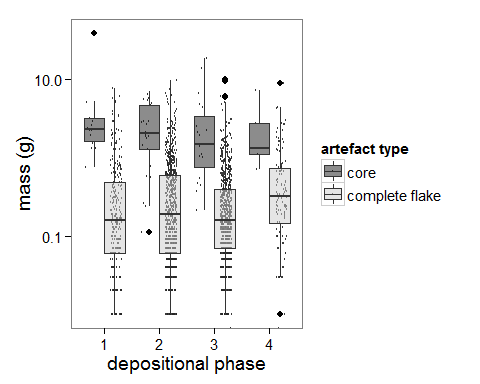
sq\_a\_cores\_mass <- sq\_a\_allchert[sq\_a\_allchert$Artclas == "core", ]  
sq\_a\_core\_metrics <- sq\_a\_cores\_mass %>%   
 group\_by(phase) %>%   
 summarise(mean(Weight),   
 mean(Length),   
 mean(Width),   
 mean(Thick),  
 sd(Weight),   
 sd(Length),   
 sd(Width),   
 sd(Thick),  
 n = length(Weight))  
sq\_a\_core\_metrics\_t <- t(round(sq\_a\_core\_metrics,2))  
colnames(sq\_a\_core\_metrics\_t) <- c("phase 1", "phase 2", "phase 3", "phase 4" )  
kable(sq\_a\_core\_metrics\_t, caption = "Square A: Summary of chert core metrics")

Square A: Summary of chert core metrics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | phase 1 | phase 2 | phase 3 | phase 4 |
| phase | 1.00 | 2.00 | 3.00 | 4.00 |
| mean(Weight) | 5.16 | 2.73 | 3.45 | 2.64 |
| mean(Length) | 22.57 | 17.89 | 17.12 | 13.27 |
| mean(Width) | 14.10 | 14.63 | 13.73 | 13.42 |
| mean(Thick) | 9.28 | 8.57 | 8.22 | 8.53 |
| sd(Weight) | 9.96 | 2.10 | 4.92 | 2.69 |
| sd(Length) | 9.07 | 5.56 | 6.65 | 2.71 |
| sd(Width) | 5.79 | 4.88 | 7.19 | 4.42 |
| sd(Thick) | 6.29 | 3.81 | 4.30 | 3.69 |
| n | 14.00 | 25.00 | 19.00 | 5.00 |

There are slightly fewer cores in the Holocene, and they tend to be slightly smaller. The code chunk below creates a figure that shows the distribution of core and complete flake mass by depositional phases

# plot flake and core mass by phase in one box plot  
sq\_a\_flakes\_cores\_weight <- sq\_a\_allchert %>%   
 filter(Artclas %in% c("flake", "core")) %>%   
 filter(Square == "A") %>%   
 select(c(Artclas, Weight, phase))  
  
ggplot(sq\_a\_flakes\_cores\_weight, aes(fill = Artclas, as.factor(phase), Weight)) +  
 geom\_point(aes(colour = Artclas), size = 0.5, alpha = 0.9, shape = 1,   
 position=position\_jitterdodge(dodge.width=0.9)) +  
 geom\_boxplot(alpha = 0.5) +  
 scale\_y\_log10() +  
 #theme\_minimal(base\_size = 4) +  
 xlab('depositional phase') +  
 ylab("mass (g)") +  
 scale\_fill\_manual(name="artefact type",  
 labels=c("core", "complete flake"),   
 values = c("grey10", "grey80")) +  
 scale\_colour\_manual(name="artefact type",  
 labels=c("core", "complete flake"),   
 values = c("grey10", "grey15"))



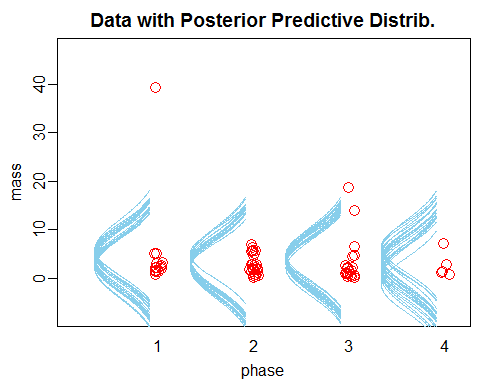
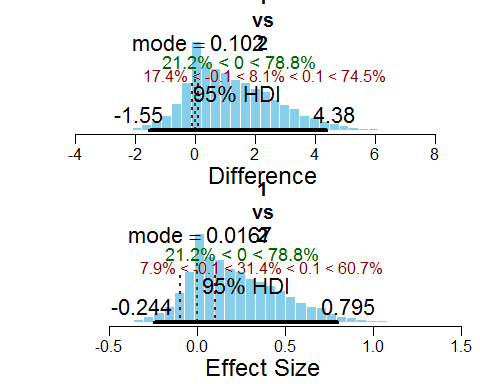
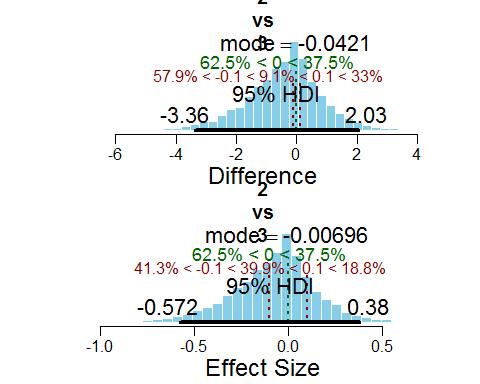
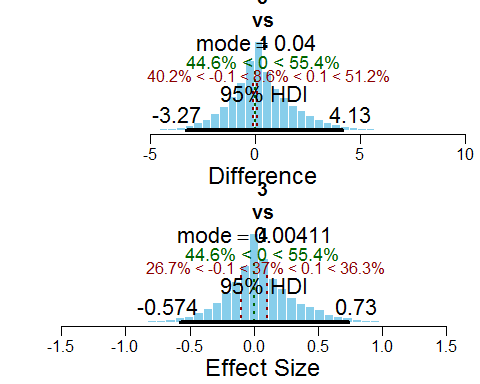
The box and whisker plot above suggests that changes in the disrtbution of core and flake mass were very subtle between the two phases.

The code chunk below uses a Bayesian one-way ANOVA to investigate differences in chert core mass by depositional phase.

myDataFrame <- data.frame(phase = sq\_a\_cores\_mass$phase, mass = sq\_a\_cores\_mass$Weight)  
yName = names(myDataFrame)[2] # mass  
xName = names(myDataFrame)[1] # phase  
contrasts = list(  
 list( c("1") , c("2") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,   
 list( c("2") , c("3") , compVal=0.0 , ROPE=c(-0.1,0.1) ),   
 list( c("3") , c("4") , compVal=0.0 , ROPE=c(-0.1,0.1) ))  
# Generate the MCMC chain:  
 mcmcCoda\_ANOVA = genMCMC\_ANOVA(datFrm=myDataFrame , yName=yName , xName=xName ,  
 numSavedSteps=11000 , thinSteps=10   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 10:10:12 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 164  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 500  
-------------------------------------------------| 500  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 1000  
-------------------------------------------------| 1000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . Updating 36670  
-------------------------------------------------| 36650  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
\* 100%  
. . . . Updating 0  
. Deleting model  
.   
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
 summaryInfo\_ANOVA = smryMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
 datFrm=myDataFrame , xName=xName ,  
 contrasts=contrasts   
 )  
   
# # Display posterior information (not easy to read unless using interactively):  
 plotMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
 datFrm=myDataFrame , yName=yName , xName=xName ,   
 contrasts=contrasts )

#   
# subset summaryInfo to show HDI interval for interactions  
HDI\_intervals\_for\_interactions <- summaryInfo\_ANOVA[c((nrow(summaryInfo\_ANOVA) - length(contrasts) + 1):nrow(summaryInfo\_ANOVA)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions,3), caption = "Square A: Highest density intervals (95%) for the posterior distributions of the interactions between phases and core mass.")

Square A: Highest density intervals (95%) for the posterior distributions of the interactions between phases and core mass.

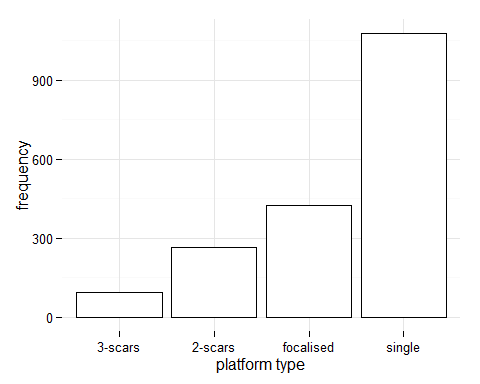
|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| 1.v.2 | -1.547 | 4.376 |
| 2.v.3 | -3.356 | 2.029 |
| 3.v.4 | -3.274 | 4.129 |

fit <- aov(mass ~ as.factor(phase), myDataFrame)  
fit\_summary <- summary(fit)

In the code chunk above we obtain the results of a frequentist ANOVA which returns a statistically non-significant result: F = 0.6040559, df = 3, p = 0.6149617. This shows a non-significant change in chert core mass. This is equivalent to what we see in the Bayesian HDI interval, which includes zero, suggesting no credible difference.

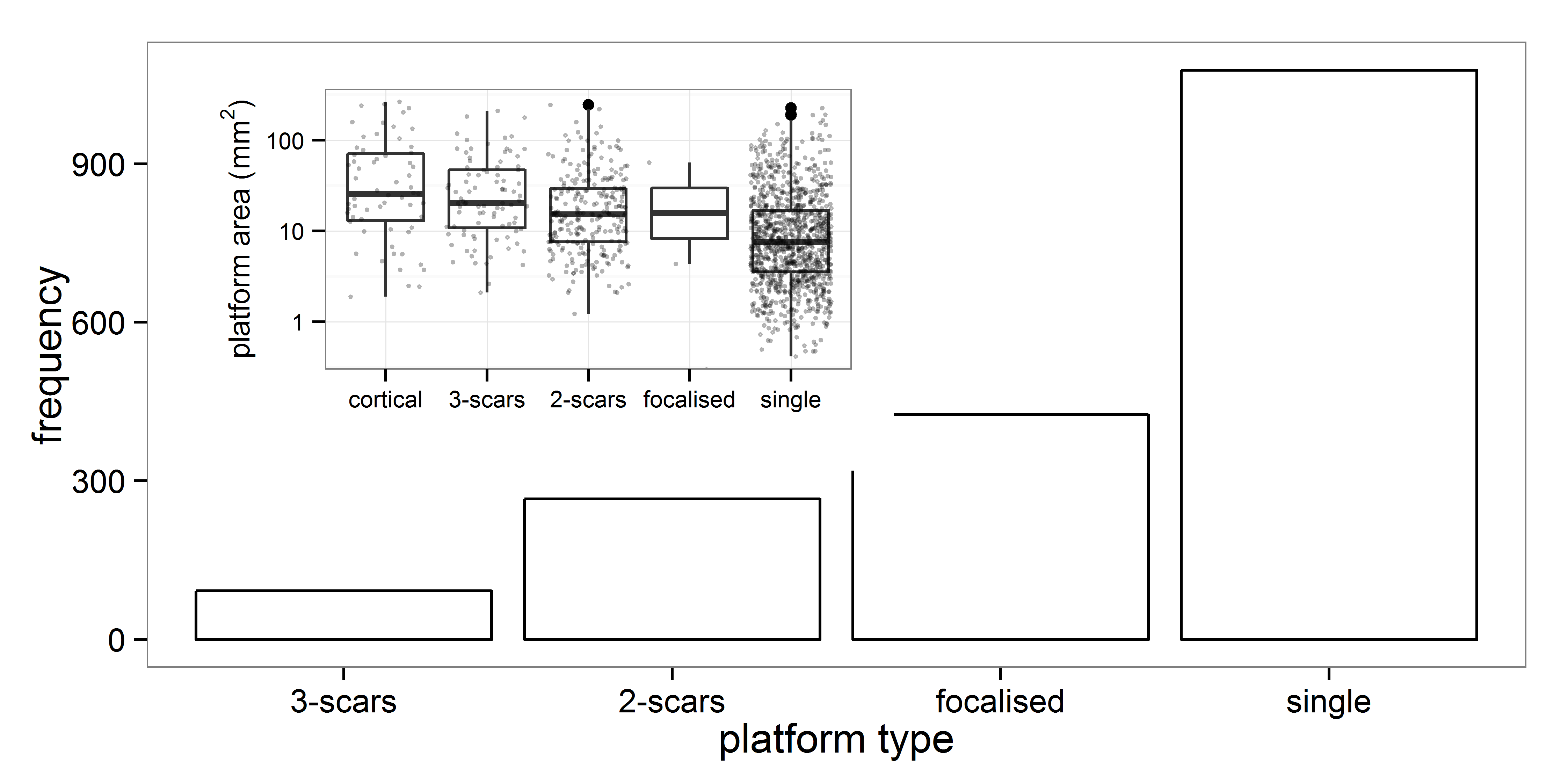
The code chunk below produces the figures that illustrate the frequency of flake platform categories for chert complete flakes at Jerimalai square A.

# flake platform  
plat <- dcast(sq\_a\_flakes, Plat ~ depth)   
rownames(plat) <- plat[,1]  
# get rid of rows with no plat  
plat <- plat[rownames(plat) != "",]  
plat <- plat[,-1]  
plat\_freqs <- data.frame(plat\_types = rownames(plat), Freq = rowSums(plat))  
plat\_freqs <- plat\_freqs[plat\_freqs$Freq > 90,]  
plat\_freqs$plat\_types <- c('2-scars', '3-scars', 'focalised', 'single')  
# plot freq of platform types  
main <- ggplot(plat\_freqs, aes(reorder(plat\_types, Freq), Freq)) +   
 geom\_bar(stat="identity", fill = "white", colour = "black") +  
 # theme\_minimal() +  
 xlab("platform type") +  
 ylab("frequency") +  
 # remove grid lines for subplot  
 theme\_update(panel.background = element\_blank(),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank())  
main + theme\_minimal()



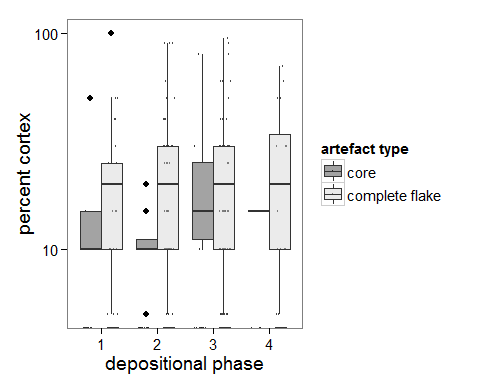
# raw materials by spit  
# compute proportions per layer (col props)  
all\_tab <- data.frame()  
for(i in seq(ncol(plat))){  
 for(j in seq(nrow(plat))){  
 all\_tab[j,i] <- plat[j,i]/colSums(plat)[i]  
 }  
}  
# check   
check <- colSums(all\_tab) # should == 1  
colnames(all\_tab) <- colnames(plat)   
check <- rowSums(all\_tab) # should be various  
all\_tab$plat\_type <- rownames(plat)   
# get rid of raw materials that are not very abundant  
all\_tab <- all\_tab[which(rowSums(all\_tab[,1:ncol(all\_tab)-1]) > 1.5) , ]  
# get rid of NA column  
all\_tab <- all\_tab[,names(all\_tab) != 'NA']  
# plot  
all\_tab\_m <- melt(all\_tab, id.var = 'plat\_type')  
# by phase  
plat <- dcast(flakes, Plat ~ phase)  
rownames(plat) <- plat[,1]  
# get rid of rows with no plat  
plat <- plat[rownames(plat) != "",]  
plat <- plat[,-1]  
# raw materials by site  
# compute proportions per phase (col props)  
all\_tab <- data.frame()  
for(i in seq(ncol(plat))){  
 for(j in seq(nrow(plat))){  
 all\_tab[j,i] <- plat[j,i]/colSums(plat)[i]  
 }  
}  
all\_tab$plat\_type <- rownames(plat)   
# get rid of raw materials that are not very abundant  
all\_tab <- all\_tab[which(rowSums(all\_tab[,1:ncol(all\_tab)-1]) > 0.15) , ]  
# get rid of NA column  
all\_tab <- all\_tab[,names(all\_tab) != 'NA']  
# plot distibution of platform sizes for each type  
sq\_a\_flakes$Platarea <- with(sq\_a\_flakes, (Platthic \* Platwid))  
plat\_area\_type <- sq\_a\_flakes[sq\_a\_flakes$Plat %in% c("Single", "Focal", "2-scars", "Cort", '3-scars'),]  
  
# make names a bit more readable  
plat\_area\_type$Plat <- ifelse(plat\_area\_type$Plat == 'Cort', 'cortical',  
 ifelse(plat\_area\_type$Plat == 'Focal', 'focalised',   
 ifelse(plat\_area\_type$Plat == 'Single', 'single', as.character(plat\_area\_type$Plat))))   
  
# put types in same order as frequency plot  
plat\_area\_type$Plat <- factor(plat\_area\_type$Plat,   
 levels = c('cortical', '3-scars','2-scars', 'focalised', 'single'), ordered = TRUE)  
# remove NA  
plat\_area\_type <- plat\_area\_type[!is.na(plat\_area\_type$Plat), ]  
   
sub <- ggplot(plat\_area\_type, aes(Plat, Platarea)) +  
 geom\_boxplot() +  
 geom\_jitter(size = 0.5, alpha = 0.3, shape = 1) +  
 scale\_y\_log10() +  
 ylab(as.expression(bquote('platform area (' \* mm^{2} \* ")" ))) +  
 xlab("") +  
 theme\_minimal()  
  
# plot freq of plat type and platform area together in one plot  
   
vp <- viewport(width = 0.45, height = 0.54,   
 x = 0.57, y = 0.4,   
 just = c("right", "bottom"))  
  
# combine plots, print and save (wont show in console)  
 png("figures/sq\_A\_Jeremalai-platform-area-by-plat-type.png",   
 units = "mm", w = 190, h = 190/2, res = 600)  
 print(main)  
 print(sub + theme\_bw(base\_size = 10), vp = vp)  
 dev.off()

png   
 2



The code chunk below creates a plot showing the distributions of core and flake cortex by depositional phase at Jerimalai square A.

sq\_a\_flakes\_cores\_cortex <- sq\_a\_allchert[(sq\_a\_allchert$Artclas == "flake" | sq\_a\_allchert$Artclas == "core"), c('Artclas', 'Cortex', 'phase') ]  
ggplot(sq\_a\_flakes\_cores\_cortex, aes( fill = Artclas, as.factor(phase), Cortex)) +  
 geom\_point(aes(colour = Artclas), size = 0.5, alpha = 0.9, shape = 1,   
 position=position\_jitterdodge(dodge.width=0.9)) +  
 geom\_boxplot(alpha = 0.4) +  
 scale\_y\_log10() +  
 xlab('depositional phase') +  
 ylab("percent cortex") +  
 scale\_fill\_manual(name="artefact type",  
 labels=c("core", "complete flake"),   
 values = c("grey10", "grey80")) +  
 scale\_colour\_manual(name="artefact type",  
 labels=c("core", "complete flake"),   
 values = c("grey10", "grey20"))



The code chunk below uses a Bayesian one-way ANOVA to investigate differences in chert flake cortex by depositional phase in square A.

myDataFrame <- data.frame(phase = sq\_a\_flakes$phase, cortex = sq\_a\_flakes$Cortex)  
yName = names(myDataFrame)[2] # cortex  
xName = names(myDataFrame)[1] # phase  
# too many zero values... makes the mode zero and breaks the model  
myDataFrame <- myDataFrame[myDataFrame$cortex != 0 & !is.na(myDataFrame$cortex), ]  
contrasts = list(  
 list( c("1") , c("2") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("2") , c("3") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("3") , c("4") , compVal=0.0 , ROPE=c(-0.1,0.1) ) )  
# Generate the MCMC chain:  
 mcmcCoda\_ANOVA = genMCMC\_ANOVA(datFrm=myDataFrame , yName=yName , xName=xName ,  
 numSavedSteps=11000 , thinSteps=10   
 )  
 # Get summary statistics of chain:  
 summaryInfo\_ANOVA = smryMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
 datFrm=myDataFrame , xName=xName ,  
 contrasts=contrasts   
 )  
 # # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
# datFrm=myDataFrame , yName=yName , xName=xName ,   
# contrasts=contrasts )  
#   
# subset summaryInfo to show HDI interval for interactions  
HDI\_intervals\_for\_interactions\_f <- summaryInfo\_ANOVA[c((nrow(summaryInfo\_ANOVA) - length(contrasts) + 1):nrow(summaryInfo\_ANOVA)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions\_f,3), caption = "Square A: Highest density intervals (95%) for the posterior distributions of the interactions between phases and flake cortex")

The code chunk above produced results that indicate that the HDI include zero, which we interpret as no credible difference. In the code chunk below we repeat the same investigation using a frequentist t-test.

myDataFrame <- data.frame(phase = sq\_a\_flakes$phase, cortex = sq\_a\_flakes$Cortex)  
fit <- aov(cortex ~ as.factor(phase), myDataFrame)  
fit\_summary <- summary(fit)

In the code chunk above we obtain the results of a frequentist t-test which returns a statistically non-significant result: F = 7.6797412, df = 3, p = 4.203331810^{-5}. This shows a non-significant change in chert flake cortex amount. This is equivalent to what we see in the Bayesian HDI interval, which includes zero, suggesting no credible difference.

In the two code chunks below we repeat the statistical tests above for core cortex.

myDataFrame <- data.frame(phase = sq\_a\_cores\_mass$phase, cortex = sq\_a\_cores\_mass$Cortex)  
myDataFrame <- myDataFrame[complete.cases(myDataFrame),] # omit NAs  
yName = names(myDataFrame)[2] # cortex  
xName = names(myDataFrame)[1] # phase  
contrasts = list(  
 list( c("1") , c("2") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("2") , c("3") , compVal=0.0 , ROPE=c(-0.1,0.1) ) ,  
 list( c("3") , c("4") , compVal=0.0 , ROPE=c(-0.1,0.1) ) )  
# Generate the MCMC chain:  
 mcmcCoda\_ANOVA = genMCMC\_ANOVA(datFrm=myDataFrame , yName=yName , xName=xName ,  
 numSavedSteps=11000 , thinSteps=10   
 )

Calling 3 simulations using the parallel method...  
Following the progress of chain 1 (the program will wait for all  
chains to finish before continuing):  
Welcome to JAGS 3.4.0 on Tue Nov 10 10:10:23 2015  
JAGS is free software and comes with ABSOLUTELY NO WARRANTY  
Loading module: basemod: ok  
Loading module: bugs: ok  
. . Reading data file data.txt  
. Compiling model graph  
 Resolving undeclared variables  
 Allocating nodes  
 Graph Size: 164  
. Reading parameter file inits1.txt  
. Initializing model  
. Adapting 500  
-------------------------------------------------| 500  
++++++++++++++++++++++++++++++++++++++++++++++++++ 100%  
Adaptation successful  
. Updating 1000  
-------------------------------------------------| 1000  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
. . . . . . Updating 36670  
-------------------------------------------------| 36650  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 100%  
\* 100%  
. . . . Updating 0  
. Deleting model  
.   
All chains have finished  
Simulation complete. Reading coda files...  
Coda files loaded successfully  
Finished running the simulation

# Get summary statistics of chain:  
 summaryInfo\_ANOVA = smryMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
 datFrm=myDataFrame , xName=xName ,  
 contrasts=contrasts   
 )  
 # # Display posterior information (not easy to read unless using interactively):  
# plotMCMC\_ANOVA( mcmcCoda\_ANOVA ,   
# datFrm=myDataFrame , yName=yName , xName=xName ,   
# contrasts=contrasts )  
#   
# subset summaryInfo to show HDI interval for interactions  
HDI\_intervals\_for\_interactions\_c <- summaryInfo\_ANOVA[c((nrow(summaryInfo\_ANOVA) - length(contrasts) + 1):nrow(summaryInfo\_ANOVA)), c("HDIlow", "HDIhigh")]  
kable(round(HDI\_intervals\_for\_interactions\_c,3), caption = "Highest density intervals (95%) for the posterior distributions of the interactions between phases and core cortex")

Highest density intervals (95%) for the posterior distributions of the interactions between phases and core cortex

|  |  |  |
| --- | --- | --- |
|  | HDIlow | HDIhigh |
| 1.v.2 | -4.229 | 8.761 |
| 2.v.3 | -9.375 | 3.189 |
| 3.v.4 | -5.393 | 11.737 |

# combine core and flake HDIs into one table  
HDIs\_for\_cores\_and\_flakes <- round(cbind(HDI\_intervals\_for\_interactions\_c, HDI\_intervals\_for\_interactions\_f),3)

The code chunk above produced results that indicate that the HDIs for all phases include zero, which we interpret as no credible difference. In the code chunk below we repeat the same investigation using a frequentist ANOVA.

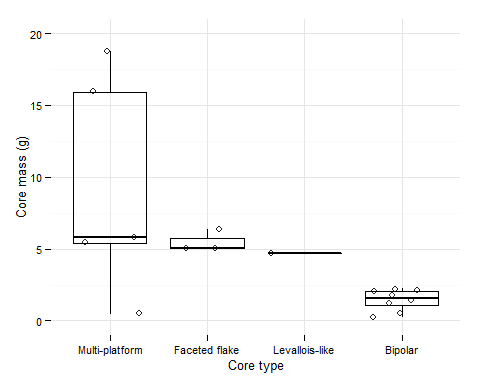
fit <- aov(cortex ~ as.factor(phase), myDataFrame)  
fit\_summary <- summary(fit)

In the code chunk above we obtain the results of a frequentist t-test which returns a statistically non-significant result: F = 0.5982667, df = 3, p = 0.6186668.. This shows a non-significant change in chert core cortex amount. This is equivalent to what we see in the Bayesian HDI interval, which includes zero, suggesting no credible difference.

## Results: Core technology

This code draws a plot of differences in the mass of cores recovered from Jerimalai by core type.

sq\_a\_core\_types <- core\_types %>% filter(Square == "A")  
sq\_a\_core\_types$Type\_long <- with(sq\_a\_core\_types, ifelse(Type == "SPC", "Single plaform",  
 ifelse(Type == "RC", "Radial",  
 ifelse(Type == "BDC", "Bidirectional",  
 ifelse(Type == "BiC", "Bipolar",  
 ifelse(Type == "MPC", "Multi-platform",  
 ifelse(Type == "LLC", "Levallois-like",  
 ifelse(Type == "FFC", "Faceted flake", NA))))))) )  
# plot  
ggplot(sq\_a\_core\_types, aes(reorder(Type\_long, -Mass, FUN=median), Mass)) +  
 geom\_jitter(alpha = 0.9, shape = 1) +   
 geom\_boxplot(alpha = 0.1, fill = "white", colour = "black") +  
 ylim(0,20) +  
 xlab("Core type") +  
 ylab("Core mass (g)") +  
 theme\_minimal(base\_size = 10)



The code chunk below computes the amount of cortex for each core type.

core\_cortex <- aggregate( X..Cortex ~ Type, data = sq\_a\_core\_types, mean)  
# what is the average amount of cortex for each core type?  
core\_cortex\_means <- arrange(core\_cortex, -X..Cortex)  
names(core\_cortex\_means) <- c("Type", "Cortex percentage")  
core\_cortex\_means[,2] <- round(core\_cortex\_means[,2],1)

Multiplatform cores (11.7%) and faceted flake cores (6.7%) exhibit the most cortex. Levallois-like cores and bipolar cores exhibit the least cortex (<4%).

The code chunk below computes the numbers of flake scars by each core type.

core\_scars <- sq\_a\_core\_types %>%  
 group\_by(Type) %>%  
 summarize(means = round(mean(Number.of.Scars, na.rm = TRUE),0),  
 sds = round(sd(Number.of.Scars, na.rm = TRUE)),0) %>%  
 arrange(means)

Levallois-like cores exhibit almost twice the number of flake scars on average as other cores in the assemblage (mean = 38 scars versus 20 on average).

## Results: Retouched artefacts

The code chunk below computes the number of retouched flakes in square A

# frequency of flakes with retouch per phase  
rt <- sq\_a\_flakes[sq\_a\_flakes$Rtch == "Yes", ]  
rt\_count <- nrow(rt)

There are only 4 retouched peices in square A, so further analysis would not help to identify patterns in retouch activity at this location.

## Results: Technological types

sq\_a\_techno\_types <- read.csv("data/Jerimalai\_technological\_table\_Square\_A.csv")  
sq\_a\_spits <- sq\_a\_techno\_types[, 1]  
sq\_a\_techno\_types[!is.na(sq\_a\_techno\_types)] <- "x"  
sq\_a\_techno\_types[is.na(sq\_a\_techno\_types)] <- ""  
sq\_a\_techno\_types$A <- sq\_a\_spits  
# split up  
sq\_a\_techno\_retouch <- sq\_a\_techno\_types[, c("End.scraper"   
 , "Side"   
 , "Scraper.Edge"   
 , "notch"   
 , "Side.and.End"   
 , "Double.Side.Scraper"   
, "Double.Side.and.End.Scraper"   
, "Denticulated..Retouch"   
, "Double.End.and.Side"   
, "Notched.double.side.and.end"   
, "Notched.side.and.end.scraper"   
, "Notched.Double.Side"   
, "Evidence.of.dual.hemispheres"   
, "drill.like.retouch")]  
sq\_a\_techno\_features <- sq\_a\_techno\_types[, c("Faceting",  
 "Brumm.and.Moore.Truncated.Flake" )]  
sq\_a\_techno\_ground <- sq\_a\_techno\_types[, c("Ground.ochre" ,  
 "Striated.haematite" )]  
sq\_a\_techno\_techtypes <- sq\_a\_techno\_types[, c("Levallois.like.flake"   
, "Burin.Spall"   
, "Burin"  
, "Bipolar.Flake"   
, "Eclat.debordant"   
, "redirecting.flake"   
, "Cortical.Flake"  
, "Faceted.Flake.from.Ventral"  
, "Unfaceted.Flake.from.Ventral"   
, "Faceted.Point")]  
sq\_a\_techno\_cores <- sq\_a\_techno\_types[, c("Truncated.faceted.core"   
, "Semi.Discoidal.Core"   
, "Flake.core.with.unfaceted.ventral.removals"  
, "Bidirectional.core"   
, "Faceted.Radial.Core.Levallois"   
, "Single.Platform.Core"   
, "Mulitplatform.Core"   
, "Bipolar.core"   
 )]  
  
sq\_a\_phases\_techo <- sq\_a\_makephases(sq\_a\_spits)  
# append phase column to each table  
techno\_tables <- list(sq\_a\_techno\_retouch, sq\_a\_techno\_features, sq\_a\_techno\_ground, sq\_a\_techno\_techtypes, sq\_a\_techno\_cores)  
techno\_tables\_out <- vector("list", length = length(techno\_tables))  
for(i in seq\_along(techno\_tables)){  
 techno\_tables[[i]]$phase <- sq\_a\_phases\_techo  
 techno\_tables[[i]][] <- lapply(techno\_tables[[i]], as.character) # change factor to character  
 techno\_tables[[i]][techno\_tables[[i]] == 'x'] <- 1 # replace x with 1  
 techno\_tables[[i]][] <- lapply(techno\_tables[[i]], as.numeric) # change char to num   
 techno\_tables\_out[[i]] <- as.data.frame(t(ddply(techno\_tables[[i]], "phase", numcolwise(sum, na.rm = TRUE))))  
 techno\_tables\_out[[i]]$type <- rownames(techno\_tables\_out[[i]])  
}  
# list to dataframe  
techno\_tables\_comb <- bind\_rows(techno\_tables\_out) # %>% View  
techno\_tables\_comb <- techno\_tables\_comb[,c(5,1,2,3,4)]  
names(techno\_tables\_comb) <- c("class", "phase 1", "phase 2", "phase 3", "phase 4")  
  
# how many spits in each phase?  
sq\_a\_spits\_per\_phase <- sq\_a\_allchert %>%   
 group\_by(phase) %>%   
 summarise(n\_spits = n\_distinct(Spit))  
  
sq\_a\_props <- lapply(seq\_along(techno\_tables\_comb)[2:5], function(i) techno\_tables\_comb[,i] / sq\_a\_spits\_per\_phase$n\_spits[i-1] )  
  
techno\_tables\_comb$`phase 1` <- sq\_a\_props[[1]]$`phase 1`  
techno\_tables\_comb$`phase 2` <- sq\_a\_props[[2]]$`phase 2`  
techno\_tables\_comb$`phase 3` <- sq\_a\_props[[3]]$`phase 3`  
techno\_tables\_comb$`phase 4` <- sq\_a\_props[[4]]$`phase 4`  
techno\_tables\_comb[2:5] <- round(techno\_tables\_comb[2:5], 2)  
kable(techno\_tables\_comb, caption = "Square A: Summary of proportions and classes. Proportions refers to the proportion of spits in each depositional phase containing a given class")

Square A: Summary of proportions and classes. Proportions refers to the proportion of spits in each depositional phase containing a given class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| class | phase 1 | phase 2 | phase 3 | phase 4 |
| phase | 0.12 | 0.15 | 0.15 | 0.8 |
| End.scraper | 0.12 | 0.23 | 0.10 | 0.0 |
| Side | 0.25 | 0.31 | 0.00 | 0.0 |
| Scraper.Edge | 0.00 | 0.15 | 0.00 | 0.0 |
| notch | 0.25 | 0.00 | 0.00 | 0.0 |
| Side.and.End | 0.00 | 0.08 | 0.00 | 0.0 |
| Double.Side.Scraper | 0.00 | 0.08 | 0.00 | 0.0 |
| Double.Side.and.End.Scraper | 0.00 | 0.15 | 0.00 | 0.0 |
| Denticulated..Retouch | 0.00 | 0.08 | 0.00 | 0.0 |
| Double.End.and.Side | 0.00 | 0.00 | 0.00 | 0.2 |
| Notched.double.side.and.end | 0.00 | 0.00 | 0.00 | 0.0 |
| Notched.side.and.end.scraper | 0.00 | 0.08 | 0.00 | 0.0 |
| Notched.Double.Side | 0.00 | 0.00 | 0.10 | 0.0 |
| Evidence.of.dual.hemispheres | 0.00 | 0.08 | 0.00 | 0.0 |
| drill.like.retouch | 0.00 | 0.00 | 0.00 | 0.0 |
| phase | 0.12 | 0.15 | 0.15 | 0.8 |
| Faceting | 0.38 | 0.62 | 0.35 | 0.0 |
| Brumm.and.Moore.Truncated.Flake | 0.12 | 0.23 | 0.00 | 0.0 |
| phase | 0.12 | 0.15 | 0.15 | 0.8 |
| Ground.ochre | 0.00 | 0.00 | 0.00 | 0.0 |
| Striated.haematite | 0.00 | 0.00 | 0.00 | 0.0 |
| phase | 0.12 | 0.15 | 0.15 | 0.8 |
| Levallois.like.flake | 0.00 | 0.00 | 0.00 | 0.0 |
| Burin.Spall | 0.12 | 0.23 | 0.05 | 0.0 |
| Burin | 0.00 | 0.00 | 0.00 | 0.0 |
| Bipolar.Flake | 0.25 | 0.46 | 0.10 | 0.0 |
| Eclat.debordant | 0.00 | 0.15 | 0.00 | 0.0 |
| redirecting.flake | 0.75 | 0.46 | 0.10 | 0.4 |
| Cortical.Flake | 0.00 | 0.00 | 0.00 | 0.0 |
| Faceted.Flake.from.Ventral | 0.25 | 0.15 | 0.15 | 0.0 |
| Unfaceted.Flake.from.Ventral | 0.50 | 0.31 | 0.35 | 0.0 |
| Faceted.Point | 0.00 | 0.00 | 0.00 | 0.0 |
| phase | 0.12 | 0.15 | 0.15 | 0.8 |
| Truncated.faceted.core | 0.00 | 0.08 | 0.00 | 0.0 |
| Semi.Discoidal.Core | 0.00 | 0.08 | 0.00 | 0.0 |
| Flake.core.with.unfaceted.ventral.removals | 0.25 | 0.00 | 0.00 | 0.0 |
| Bidirectional.core | 0.00 | 0.00 | 0.00 | 0.0 |
| Faceted.Radial.Core.Levallois | 0.00 | 0.00 | 0.00 | 0.0 |
| Single.Platform.Core | 0.00 | 0.00 | 0.00 | 0.0 |
| Mulitplatform.Core | 0.12 | 0.23 | 0.05 | 0.0 |
| Bipolar.core | 0.12 | 0.23 | 0.10 | 0.0 |

# Colophon

This report was generated on 2015-11-10 12:22:39 using the following computational environment and dependences:

# which R packages and versions?  
sessionInfo()

R version 3.2.2 (2015-08-14)  
Platform: x86\_64-w64-mingw32/x64 (64-bit)  
Running under: Windows 7 x64 (build 7601) Service Pack 1  
  
locale:  
[1] LC\_COLLATE=English\_United States.1252   
[2] LC\_CTYPE=English\_United States.1252   
[3] LC\_MONETARY=English\_United States.1252  
[4] LC\_NUMERIC=C   
[5] LC\_TIME=English\_United States.1252   
  
attached base packages:  
[1] grid stats graphics grDevices utils datasets methods   
[8] base   
  
other attached packages:  
 [1] JerimalaiStoneArtefacts\_0.0.0.9000 Bchron\_4.1.1   
 [3] inline\_0.3.14 dependencies\_0.0-1   
 [5] git2r\_0.10.1.9000 data.table\_1.9.4   
 [7] xtable\_1.7-4 lattice\_0.20-33   
 [9] BEST\_0.3.0 runjags\_2.0.1-4   
[11] rjags\_3-15 coda\_0.17-1   
[13] vcd\_1.4-1 dplyr\_0.4.3   
[15] plyr\_1.8.3 reshape2\_1.4.1   
[17] ggplot2\_1.0.1 rmarkdown\_0.8.1   
[19] printr\_0.0.4 knitr\_1.11   
  
loaded via a namespace (and not attached):  
 [1] Rcpp\_0.12.1 highr\_0.5.1 formatR\_1.2.1 tools\_3.2.2   
 [5] mclust\_5.0.2 digest\_0.6.8 jsonlite\_0.9.16 evaluate\_0.8   
 [9] gtable\_0.1.2 rstudioapi\_0.3.1 DBI\_0.3.1 jagsUI\_1.3.7   
[13] yaml\_2.1.13 parallel\_3.2.2 proto\_0.3-10 stringr\_1.0.0   
[17] lmtest\_0.9-34 ellipse\_0.3-8 hdrcde\_3.1 R6\_2.1.1   
[21] magrittr\_1.5 codetools\_0.2-14 scales\_0.3.0 htmltools\_0.2.6   
[25] MASS\_7.3-43 assertthat\_0.1 colorspace\_1.2-6 labeling\_0.3   
[29] stringi\_0.5-5 lazyeval\_0.1.10 munsell\_0.4.2 chron\_2.3-47   
[33] zoo\_1.7-12

# what other pieces of software?  
needs <- needs()  
c(needs$depends$SystemRequirements[needs$depends$SystemRequirements != "NULL"], needs$imports$SystemRequirements[needs$imports$SystemRequirements != "NULL"])

[1] "zlib headers and library. OpenSSL (non-Windows)\nheaders and library. Optional LibSSH2 (non-Windows) to enable the\nSSH transport."  
[2] "JAGS (http://mcmc-jags.sourceforge.net)"   
[3] "JAGS (>= 3.0.0) (see\nhttp://mcmc-jags.sourceforge.net)"   
[4] "pandoc (>= 1.12.3) -\nhttp://johnmacfarlane.net/pandoc"   
[5] "JAGS (http://mcmc-jags.sourceforge.net)"   
[6] "ICU4C (>= 50, optional)"

# what commit is this file at?  
repo <- repository(path = "../")  
last\_commit <- commits(repo)[[1]]

The current git commit of this file is 7a43033ead0a2d617a0c0422f84f5b6bfb495e6a, which is on the master branch and was made by rstudio on 2015-11-10 02:20:26. The current commit message is "update sq A phases".