The utility of statistical analysis in structural geology

Nicolas M. Roberts[1*], Basil Tikoff[1], Joshua R. Davis[2], Tor Stetson-Lee[1]

[1]Department of Geoscience, University of Wisconsin—Madison. 1215 West Dayton St., Madison, WI 53715*

[2]Department of Mathematics and Statistics, Department of Computer Science, Carleton College, Northfield, MN

* nmroberts@wisc.edu

* *(608) 262-4678*

Key words: structural geology; directional statistics; orientation statistics; western Idaho shear zone; inference; regressions

Abstract

Recent advances in statistical methods for structural geology make it possible to treat nearly all types structural geology field data. These methods provide a way to objectively test hypotheses and to quantify uncertainty, and their adoption into standard practice is paramount to the future of structural geology. We provide a blueprint for structural geologists seeking to incorporate statistics into their workflow through statistical analyses in two locations within the western Idaho shear zone. In the N-S West Mountain location, we test the published interpretation that there is a bend in shear zone at the kilometer scale. Directional statistics on foliations corroborate this interpretation, while orientation statistics on foliation-lineation pairs do not. This discrepancy leads us to reconsider an assumption made in the earlier work. In the NW-SE Ahsahka location, we present results from a full statistical analysis of foliation-lineation pairs, including data visualization, regressions, and inference. These results agree with thermochronological

evidence that suggests that the Ahsahka segment comprises two distinct, subparallel shear zones. The R scripts that were used for both statistical analyses can be found in the appendices, with the intention that they be downloaded and run alongside the results section of this paper.

1. Introduction

Structural geologists routinely work with datasets that are logistically limited to small sample size and/or spatial extent. When working with such data, an important—but under-appreciated—task should be to determine what can reasonably be interpreted about the geologic system in question. This determination depends on the uncertainty that arises because the dataset is an incomplete representation of the larger system. The field of statistics is fundamentally concerned with this data-to-system uncertainty, and statistical methods have important utility for any empirical research. As structural geologists, we can use statistics to better identify trends, understand mean(s) and dispersion in datasets, test hypotheses, evaluate implicit assumptions, and communicate the confidence of our interpretations to peers.

In most publications, structural geologists make interpretations using quantitative data (e.g. fabric measurements) and qualitative estimates of uncertainty. The lack of statistical treatment of structural geology data is in part a historical issue: there is not a strong tradition of training structural geologists in statistics. As a discipline born out of field studies and geologic mapping, early structural geology methods—including quantitative ones—developed without a statistical framework. Even with the eventual development of such a framework for *directional* (rays and lines) data types such as paleomagnetic poles, lineation, pole to foliation, paleocurrent, and fault striations (e.g. Davis and Sampson, 1986; Ducharme et al., 1985; Fisher et al., 1989; Jupp and Mardia, 1989; Merrett and Allmendinger, 1990; Yonkee and Weil, 2015), the statistically savvy structural geologist is still unusual. Though contouring directional data has become commonplace thanks to computer programs such as Stereonet (version 10.0.0; Allmendinger, 2017) and Orient (version 3.6.3; Vollmer, 2017), structural geologists do not generally make use of directional statistics to report statistical descriptors (mean, dispersion) or perform hypothesis tests in a statistically rigorous fashion.

Another reason structural geologists do not generally employ statistics is that many geologic data are not rays or lines, and thus cannot be treated with directional statistics. Until recently, there was no unified framework for the statistical treatment of *orientation* (line-in-plane) data like foliation-lineation pairs, fault planes with slickenlines, axial planes with hinges, or focal mechanisms. Davis and Titus (2017) have developed the mathematical background and theory of orientation statistics in order for it to be accessible by structural geologists. Moreover, they developed a free R programming language library for both direction and orientation statistics, called *geologyGeometry* (download at: http://www.joshuadavis.us/software/). Tools in the library include advanced plotting, regression algorithms, and parametric and non-parametric methods for inference, including hypothesis testing. The *geologyGeometry* library calls on other R libraries, including *Directional* for directional statistics (Tsagris and Athineou, 2016).

This contribution describes how to perform statistical analysis on structural geology data, and illustrates why incorporating statistics into a structural geology workflow is critical to the future of structural geology. Statistical analysis of two datasets from the western Idaho shear zone system, Idaho, USA are described in detail. The datasets were chosen because they address common questions in structural geology. A cursory geologic context is provided for each dataset (see Appendix 1 for additional details). The analysis of these two datasets reveals how thinking statistically leads to a more objective approach to interpretations and a quantified understanding of the uncertainty surrounding these interpretations. By demonstrating both its methodology and utility, we hope to motivate the adoption of statistics into the standard structural geology workflow. All statistical analyses were done with the *geologyGeometry* R library, and the full analyses are shared in the appendices. Readers are encouraged to download the *geologyGeometry* R library and run the scripts found in the appendices line by line as they read. The scripts will output interactive, rotatable versions of many of the plots found in the figures.

2. The statistical approach

In the statistical analysis applied to each dataset in this paper, a new workflow motivated by both statistical protocol and geologic expertise has been applied. This workflow highlights two types of questions in statistics that are particularly relevant for structural geologists. First, are two datasets or subsets of a single dataset (e.g. two geographic domains) sampled from the same population? Second, are there real, systematic trends in the data, based on geographic position or any other variable?

Figure 1 presents a diagram of this workflow. First, the structural geologist visualizes the data in a variety of plots and maps. As a result of visualization, the geologist makes hypotheses and qualitative interpretations about the geologic system under study (lower path in Fig. 1), from which possible conceptual models and predictions are developed. Simultaneously, a statistical protocol is executed that is necessary for any dataset (upper path). Model predictions may be objectively tested by regressions (grey arrows), in which case the upper and lower path overlap. The data can be statistically described by mean and dispersion, and if there are no systematic spatial tendencies in the data, then inferences can be made about how well the population mean is known (the uncertainty of the population mean). The upper and lower path interact again when model predictions are tested using this uncertainty in statistical hypothesis testing. A statistical hypothesis is formulated as a null hypothesis (e.g., "There is no difference in the mean of the populations from which dataset *A* and dataset *B* were sampled") and an alternative hypothesis (e.g., "The means of the populations from which dataset *A* and dataset *B* were sampled do not have the same mean"). The null hypothesis is rejected or fails to be rejected based upon a credibility or confidence threshold, commonly 95%, or a *p*-value threshold, usually $p < 0.05$. A rejection of the null hypothesis leads a structural geologist to conclude that dataset *A* and dataset *B* are sampled from different populations. Importantly, though, the failure to reject the null hypothesis would *not* lead a structural geologist to conclude that dataset *A* and dataset *B* come from the same population—only that there is not strong evidence that they came from different populations.

Interpretations about the geologic system all pass through the statistical portion of the workflow, but importantly statistics are only useful insofar as they complement geologic expertise to develop statistically constrained conclusions about the geologic system. The first example in this paper focuses on the "Mean, dispersion," "Inference about the mean," and "Statistical hypothesis testing" boxes of the statistical workflow (Fig. 1) to statistically test a published geologic interpretation. The second example illustrates a path through the workflow (as shown by railroad ties on arrows in Fig. 1).

3. Background: direction and orientation data

The mathematical description of a geologic structure's orientation depends on the type of geologic structure. A lineation can be described by two angles, a trend and plunge, or by a vector in Cartesian (north, east, up) coordinates. A foliation can also be described by two angles—a strike and a dip—and similarly can be described as a single Cartesian vector that defines the pole to the foliation. Foliations and lineations are both examples of *directional* data, meaning that a single line or ray is sufficient to uniquely describe the geometry. A wealth of statistical techniques have been developed for directional data (e.g. Mardia and Jupp, 2000), termed *directional statistics*.

In contrast, a foliation-lineation pair is a line within a plane. At a minimum, three angles are required to describe a unique foliation-lineation: a strike, a dip, and a rake. Foliation-lineation pairs are an example of *orientation* data, and are treatable by *orientation statistics* (Davis and Titus, 2017). For statistical treatment, orientation data can be represented by a 3 by 3 rotation matrix, with the first row comprising the Cartesian vector of the pole to foliation, the second row comprising the vector of the lineation, and the third row comprising the vector that is orthogonal to the first two rows (Davis and Titus, 2017).

4. Application 1: The western Idaho shear zone near West Mountain, ID

The western Idaho shear zone forms a steep and abrupt north-south boundary between accreted terranes and the cratonic edge of the North American Cordillera (Armstrong et al., 1977; Fleck and Criss, 1985; Manduca et al., 1992; Fleck and Criss, 2004; Braudy et al., 2017). The ~5 kilometer wide shear zone is characterized by highly deformed orthogneisses. In the West Mountain area, the western Idaho shear zone is dextral, but sub-vertical lineations suggest transpression (Giorgis et al., 2008; Giorgis et al., 2016). The shear zone system bends at the 100-km scale to follow the cratonic boundary as defined by the $^{87}Sr/^{86}Sr$ isopleth.

A recent structural study suggests that a subtle bend in shear zone orientation can also be detected at the kilometer scale near West Mountain, ID. Braudy et al. (2017) collected a dataset of field fabrics including both foliation-only measurements as well as foliation-lineation pairs (Fig. 2). They plot both types of fabric data in equal area nets and interpret a ~20° rotation between foliation strike in the North and South of the field area. In this section, we demonstrate a statistical analysis that is able to provide a more objective test of the interpretation of Braudy et al. (2017). In addition, we test the assumption that the foliations from the foliation-only dataset and the foliation-lineation dataset are the same. Foliation-only and foliation-lineation pairs are treated separately because they are different data types.

4.1 Directional statistics on foliation-only data

Foliation-only data comprise 148 field fabric measurements. Braudy et al. (2017) divide these data into three geographic domains: northern (n = 56), central (n = 23), and southern (n = 69) (Fig. 2B). For the sake of comparison, these domain divisions are used in this statistical analysis.

For each domain, the data are used to make an inference about the population mean. This calculation is done by bootstrapping (Efron and Tibshrani, 1994) and by applying three two-sample tests for comparison. Bootstrapping is repeated sampling with replacement. The implementation of the bootstrapping routine in the *geologyGeometry* R library is straightforward to use and automatically computes confidence regions (see Appendix 2 for a simplified description, or Davis and Titus (2017) for a full description). The result of bootstrapping is a cloud of means, whose center approximates the mean of the dataset and whose density at any given point is related to the likelihood of that point being the population mean. The 95% confidence ellipse of each domain is calculated using the Mahalonobis distance (Mahalanobis, 1936).

To determine whether the foliations in each domain come from different populations, a series of statistical tests are devised. The null hypotheses are that the population mean of one domain (e.g. northern) is the population mean of another domain (e.g. southern). The null hypothesis is rejected if the 95% confidence regions of the two domains in question do not overlap.

Results of bootstrapping and 95% confidence region calculations are summarized in Figure 3A. The 95% confidence ellipses of the southern and central domains overlap, but neither of these domains overlap with the northern region. Therefore, we fail to reject the null hypothesis that the southern and central domains are sampled from the same population, while for comparisons with the northern domain, we reject the null hypothesis of a single population at 95% confidence.

In addition to bootstrapping, three types of two-sample tests were applied to each pair of domains. See Appendix 2 for a brief description of each test. A Wellner test (Wellner, 1979) yields a $p < 0.0001$ based on 10,000 permutations for the northern and southern domains as well as for the northern and central domains. The p-value for the southern and central comparison is 0.427. Two variations of Watson inference tests were performed on the data (Mardia and Jupp, 2000), one that assumes tightly concentrated data and the other that assumes large sample size. Respectively, these tests yielded p-values of 0.000001 and 0 (northern and southern domains), 0.00002 and 0.000005 (northern and central domains), and 0.233 and 0.131 (central and southern domains). These p-values agree well with the bootstrapping results, although some caution is advised, since these tests make assumptions about how the data are distributed. Taken together, these tests provide strong evidence against the null hypotheses that the northern domain and the central/southern domains are sampled from the same population.

Given the statistically significant difference between the northern domain and the other two domains, we calculate the rotational difference between the northern and southern domains. The axis and magnitude of rotation between the northern and southern domains is determined by computing the minimum rotations between 10,000 pairs of northern and southern bootstrap means.

Results from this analysis are summarized in Figure 3A. The mean rotation is 12.20° ± 3.8° (2σ) with a mean rotation axis that trends 164.3° and plunges 68.8°. These results contrast the interpretation of Braudy et al. (2017), who suggested a ~20° rotation and implicitly assumed a vertical axis rotation.

4.2 Orientation statistics on foliation-lineation data

Foliation-lineation data comprise 129 field fabric measurements. The data are analyzed with the same geographic domains described above: northern ($n = 16$), central ($n = 34$), and southern ($n = 79$).

For each domain, both a bootstrapping method and a Markov chain Monte Carlo (MCMC) simulation (Davis and Titus, 2017) produce clouds of possible means from which a confidence region (for bootstrapping) or a credible region (for MCMC) can be computed. See Appendix 2 for a brief description. In general, for small sample sizes ($n < 30$), MCMC returns credible regions with accurate size, but the regions tend to be unrealistically isotropic. By contrast, bootstrapping returns more realistic anisotropic confidence regions, but the size of the region is consistently underestimated. Because of these complementary strengths and weaknesses, it is helpful to use both approaches.

The null hypotheses for foliation-lineation pairs are identical to those for foliations described previously. If the bootstrap/MCMC confidence/credible regions do not overlap, then the null hypothesis can be rejected at 95% confidence/credibility.

Figure 3B shows the results of both bootstrapping and MCMC. The null hypothesis that the northern and central domains are sampled from populations with the same mean cannot be rejected using MCMC, but can be rejected using bootstrapping at 95% confidence. The same is true for the null hypothesis with respect to the northern and southern domains. The null hypothesis that the southern and central domains are sampled from populations with the same mean cannot be rejected at 95% confidence/credibility. Because the MCMC method tends to have more accurate credibility than bootstrapping confidence intervals, these analyses do not provide strong evidence that differences among the three domains are statistically significant, although there is weak evidence that the null hypothesis can be rejected for the northern domain with respect to the other two domains.

4.3 Comparing foliation-only and foliation-lineation data

The statistical analysis of foliation-only and foliation-lineation fabric data from Braudy et al. (2017) leads to two different interpretations. From the foliation-only data, a 12.55° ± 3.30° (2σ) rotation between the southern/central and northern domains is inferred. From the foliation-lineation data, no such difference can be inferred with statistical significance. This discrepancy motivates a statistical comparison of these two datasets.

In a final comparison, only the foliations are used from the foliation-lineation data so that directional statistics can be applied to both datasets. The null hypothesis for each domain is that the foliations from the foliation-lineation dataset are sampled from the same population as those from the foliation-only dataset. A comparison of the bootstrapped mean cloud for each domain (Fig. 3C) shows that the null hypothesis can be rejected with 95% confidence for the central and southern domains, but is not clearly rejected for the northern domain. This result is unexpected because foliation-only data and foliation-lineation data were collected in the same field area (similar extent and spacing) and were assumed to be the same (Fig. 2A).

4.4 Summary

The statistical analysis of foliation and foliation-lineation data in the West Mountain area of the western Idaho shear zone allows for interpretations with quantitative evaluation of uncertainty. In addition, statistical comparison between foliation-only and foliation-lineation data reveals that a basic assumption about the two datasets—that the foliation and foliation-lineation datasets are being sampled from the same population—may not be valid.

The interpretation that Braudy et al. (2017) make with respect to foliation-only differences between the southern/central and northern domains is reasonable. Statistical analysis shows that a rotation is statistically significant. The magnitude of that rotation, however, is likely smaller than Braudy et al. (2017) interpreted, even though both this paper and the earlier study assume the smallest rotation is the geologically real one. Further, the axis of shortest rotation is not vertical, as implicitly presumed in Braudy et al. (2017), but plunges steeply to the south.

Statistical analysis of foliation-lineation pairs does not corroborate the counterclockwise rotation of fabric from south to north proposed by Braudy et al. (2017). The difference in fabric orientation among the domains is not statistically significant.

Braudy et al. (2017) do not interpret foliation-lineation pairs independently of the foliation-only data. By combining the datasets, they assume that foliations from the two datasets were sampled from the same population within each domain. Three statistical comparisons of the foliation data from the two datasets within each domain reject this assumption with 95% confidence for the central and southern domains. All three domains of the foliation-lineation foliations plot in the gap between the northern and southern domains of the foliation-only foliations (Fig. 3C). There are several possible explanations for this discrepancy which motivate future work. One possibility is that LS-tectonites may have a different orientation than S-tectonites because of strain partitioning within the western Idaho shear zone. Another possibility is that rocks that had only foliations also have weaker fabric, which may account for the larger spread of data. Whatever the case, new scientific questions arise from the statistical analysis that would not have been asked in the absence of statistical methods.

5. Application 2: Ahsahka and Woodrat Mountain shear zones near Orofino, ID

The Ahsahka shear zone is in structural continuity with the western Idaho shear zone, about 200 km north of the West Mountain area. The Ahsahka shear zone occurs within a 90° bend of the western Idaho shear zone system (e.g., Lewis et al., 2014). The current interpretation is that there is an older, parallel Woodrat Mountain shear zone in cryptic contact with the northeast boundary of the Ahsahka shear zone (Lewis et al., 2014, Schmidt et al., 2016), but this interpretation is based upon relatively little geochronological data and remains controversial.

A recent dataset collected by Stetson-Lee (2015) comprises foliation-lineation measurements from areas on either side of the cryptic boundary between what is currently mapped as the Ahsahka and Woodrat Mountain shear zones near Orofino, Idaho (Fig. 4). There is a generally NW-striking foliation throughout the field area. Cooling $^{40}Ar/^{39}Ar$ ages on hornblende, biotite, and muscovite suggest that rocks on either side of the inferred boundary between the Ahsahka and Woodrat Mountain shear zones have a protracted thermal history, and record at least two distinct events (Davidson, 1990).

The goal of this statistical analysis is to assess whether structural data support the current interpretation of two shear zones. If geographic domains have statistically significant orientation differences, are these differences consistent with the current inferred boundary? This statistical analysis illustrates the proposed structural geology workflow (Fig. 1), with a particular emphasis on the “statistical protocol” path. First, the data are visualized through a variety of plots. Second, the data are tested for geographic trends using regressions, and are split into geographic domains as a result. Third, the domains are statistically described with mean and dispersion. Finally, the domains are compared using hypothesis testing. The statistical tests are motivated and informed by data from the literature (maps, cooling ages) and the conceptual model for shear zone boundary that arises from them.

5.1 Statistical analysis using orientation statistics

The Orofino dataset comprises 69 foliation-lineation pairs in three geographic areas: Domain 1 (n = 23), domain 2 (n = 14), and domain 3 (n = 32) (Fig. 5). The division of the data in this way is consistent with current interpretations of geologic boundaries and will be statistically tested.

Initial plots contain all the data, not yet divided into geographic domains (Fig. 5A). Equal-area net and equal-volume plots with Kamb contours show that the foliation-lineation data have an approximately unimodal distribution in their orientation. However, coloring the data by geographic location reveals a non-random relationship between orientation and geography. For example, when the data are color-coded by northing, there are clear domains of yellows and reds (Fig. 5A). In map view, this geographic dependence is apparent; measurements in the south have a north to northwest lineation trend, while in the north, the lineations trend northeast to east.

Before splitting the data into domains, it is critical to know whether this geographic dependency is systematic (i.e. can be described by a continuous function) or whether there are discrete differences of orientation in different geographic domains. A series of 18 geodesic regressions help answer this question (Fig. 5b). Each of these regressions fits a geodesic curve to the data as a function of an azimuth (e.g. northing). The maximum R^2 of a geodesic regression is 0.13 (for an azimuth of 30°). A kernel regression, which fits a more complex function to the data, of 30° azimuth has an R^2 value of 0.522.

These low R^2 values suggest that the geographic dependency observed in the equal area and equal volume plots is probably not systematic, and leads to the division of the data into multiple domains (Fig. 5C). The same plotting and regression analysis for each domain suggests that there is no strong geographic dependence (see Appendix 6).

Within each domain, the data are approximately unimodal and symmetric about that mode, so the mean is an appropriate summary statistic. We use the Fréchet mean which is the point that minimizes the Fréchet variance (Table 1). The dispersion of the data can be described using the matrix Fisher maximum likelihood estimation. This dispersion measure is not meaningful geologically, but is critical to selecting which inference method is most appropriate (Davis and Titus, 2017). In this case, MCMC simulation is the best behaved. As a check, bootstrapping has also been done.

MCMC and bootstrapping results for each domain are shown in Fig. 5D and 5E, with 95% credible/confidence ellipsoids. The null hypotheses are that each pair of domains are sampled from populations with the same mean. The credible/confidence regions of domain 2 and 3 overlap appreciably, while the credible/confidence region of domain 1 does not overlap with the other two. The null hypothesis that domains 2 and 3 are sampled from populations with the same mean cannot be rejected. The null hypothesis that domain 1 and domain 3 are sampled from populations with the same mean can be rejected with 95% credibility/confidence. The null hypothesis for domains 1 and 2 can also be rejected with 95% credibility/confidence.

5.2 Summary

There are three first-order conclusions that can be drawn from the statistical analysis of the foliation-lineation pairs near Orofino, ID. First, there are geographic domains; within each domain the data are roughly unimodal and symmetric, and apparent spatial dependencies have consistently low R^2 values. Second, the difference between orientations of domains 2 and 3 is not statistically significant. Third, domain 1 is significantly different from the other two domains.

These results are consistent with the mapped boundary between the Ahsahka and Woodrat Mountain shear zones as defined by cooling ages. Domains 2 and 3 are along strike of one another, and have been previously interpreted to be part of the Woodrat Mountain shear zone. Domain 1, which is across strike from the other two domains, has been interpreted to be part of the later Ahsahka shear zone. The presence of distinct orientations in rocks with different $^{40}Ar/^{39}Ar$ cooling ages, confirmed to be statistically significant in this analysis, provides further evidence that there were two distinct shear zones, now located adjacent to one another.

6. Discussion

In both the West Mountain and Orofino datasets, a workflow that incorporates statistical analysis leads to interpretations that are tested in an objective way with reported uncertainty—some of which would not likely not have been made otherwise. At West Mountain, we show that foliation-lineation pairs are not statistically distinguishable in different geographic areas, and that foliations in the foliation-only dataset are demonstrably different from foliations in the foliation-lineation dataset. In addition, we computed the magnitude of rotation between the northern and southern domains in a way that incorporates the uncertainty about the mean of each domain. In the Orofino area, we show that the difference in orientation on opposite ends of the Dworshak

Reservoir could not be accounted for by systematic spatial variations and that the difference in orientation between the southwest shore of the reservoir (domain 1) and the northeast shore (domains 2 and 3) is statistically significant. This division is consistent with other geologic data.

The statistical approach has tangible scientific benefits for structural geology data. Statistical methods help to quantitatively identify spatial tendencies in high-dimensional data. They also can be used to quickly compute basic statistical descriptors of the dataset and uncertainty about the population mean, which helps guide the visualization of data. The uncertainty of the population mean is used to reject (or fail to reject) geologic hypotheses posed as statistical hypotheses; using this approach to testing hypotheses, geologic interpretations come with quantifiable uncertainties which structural geologists can report in publications. The statistical approach also allows structural geologists to assess the validity of implicit assumptions using the same methods that are used to test geologic hypotheses.

6.1 Identifying spatial tendencies

In our regressions of foliation-lineation data, we treat each data point holistically as a rotation matrix. This approach is an improvement on standard practice in structural geology. The investigation of geographic trends in structural geology data usually involves the decomposition of high-dimensional data like foliation-lineation pairs into one-dimensional elements. For example, it is common to plot the strike of foliation against the distance from a shear zone, even though each data point is a strike, dip, and rake. These two-dimensional charts have some utility, but provide an incomplete view of each data point. Best-fit lines and associated R^2 values in these charts are problematic because such regressions should be informed by the other dimensions that comprise each data point. This partial view of the data can lead to false correlations or can fail to reveal correlations entirely. By treating the data holistically during statistical analysis, structural geologists can be more accurate in their identification of spatial dependencies.

Performing regressions is also a critical step when testing that the data are spatially independent. Common techniques for inference about the mean of a population assume this independence. An iterative process of visual inspection (plotting), regression analysis, and division of the data into domains (as performed in the Orofino example) ensures that this assumption is reasonable.

6.2 Uncertainty about the mean

Structural geology datasets are often relatively small and dispersed. This situation is especially true for field datasets such as the ones described in this paper. Equal-area projection computer programs widely used by structural geologists have built-in measures of mean and dispersion for directional data (e.g. foliations only), but do not yet treat orientation data (e.g. foliation-lineation pairs). For foliation-lineation data, we employ one of two equally valid conceptions of the mean (Davis and Titus, 2017). To quantify how well the mean of the dataset reflects the mean of the population from which the dataset was sampled, bootstrapping and MCMC simulations produce clouds of means from which confidence and credibility regions can be inferred. The confidence/credible region for the mean of a structural geology dataset has two main functions. First, it contextualizes the mean of the dataset—a mean is not particularly useful if the uncertainty about that mean is very large. Second, confidence/credible regions enable comparison with other datasets using hypothesis testing.

6.3 Hypothesis testing

In both examples provided in this paper, an experienced structural geologist would most likely notice differences among some of the domains. Taking a statistical approach, structural geologists can test hypothesized differences in an objective way. In practice, if the 95% confidence/credible regions of two domains do not overlap, the null hypothesis that they are sampled from the same population can be rejected.

Statistical significance is especially important when the difference between two datasets is small or data are dispersed. In the West Mountain example, foliation-only foliations and foliation-lineation foliations plot in the same general area of the equal area net. While visual inspection may lead a structural geologist to suspect a difference between the two datasets, it is only through statistical hypothesis testing that the geologist can say with 95% confidence that this difference is not due to random variation within the same population. We are able to rely on this interpretation to ask further questions—such as why foliation-lineation pairs are different from foliation-only data—precisely because we have rejected the null hypothesis that they come from the same population.

6.4 Using hypothesis testing and regressions to assess assumptions

A statistical comparison of the two West Mountain datasets led to the conclusion that foliations from foliation-only data were likely not sampled from the same population as those from foliation-lineation data. This finding leads us to reject an assumption that Braudy et al. (2017) made which seemed logical. The ability to quickly assess such assumptions is a key advantage of the statistical workflow. In part, this advantage comes from a shift in perspective, because the statistical approach forces the articulation (and thus awareness) of the assumptions we make when analyzing data.

6.5 Better science through statistics

The use of statistics in structural geology may seem onerous, simply another task to complete prior to submitting a manuscript. However, given the examples above we suggest that there are many reasons to adopt this methodology. Taken together, the benefits of the statistical approach make it easier to have the scientific integrity Feynman (1974) discussed in his famous essay "Cargo Cult Science":

> *The first principle is that you must not fool yourself—and you are the easiest person to fool. So you have to be very careful about that. After you've not fooled yourself, it's easy not to fool other scientists. You just have to be honest in a conventional way after that.*

When conducting fieldwork, many (if not most) of the hypotheses that we initially formulate are ultimately incorrect. The successful execution of science is the ability to generate and discard hypotheses with relative efficiency. Statistics aids in perhaps the most difficult part of the scientific process: exactly when to discard a hypothesis. Statistical analysis is used by most scientific communities—including field scientists such as ecologists—to facilitate this process. As Henry Pollack puts it in *Uncertain Science, Uncertain World,* the role of science is to separate the "demonstrably false from the probably true" (Pollack, 2003), and the uncertainty computed by statistical methods is a primary tool to fulfill that role. The relatively small size and large dispersion common to structural geology datasets may seem a good excuse not to use statistics. In fact, these characteristics are particularly compelling reasons to incorporate statistical tools into the structural geology workflow. It can be tempting to over-interpret small datasets, and statistics provides a check on what interpretations are permissible given the small sample size.

Finally, the institution of science depends on the presentation of data and interpretations to scientific peers. Most data in structural geology papers present data in the form of stereonets or other representative documentation, neither of which allows other structural geologists to evaluate or use the dataset effectively. A clear statement of the tested hypotheses and the results would be a useful way to communicate the uncertainty of the data with respect to a specific model. The theory for the types of data that we collect has been addressed by Davis and Titus (2017) and the tools to statistically analyze data are now easily available. With the addition of specific use cases—as introduced in this contribution—we hope that both the methodology and its utility will be clear and accessible to the structural geology community.

Conclusion

Structural geology, especially structural geology in the field, is a science that benefits from the incorporation of statistical procedures. Field datasets are commonly small, geographically dispersed, and limited to small areas of good outcrop. Further, structural geology data are inherently high-dimensional, meaning that traditional ways of viewing data provide incomplete pictures of the data. The analysis of structural geology data within a statistical framework provides a way for structural geologists to more quantitatively understand and interrogate their data.

In this contribution, two typical structural geology field datasets were analyzed using direction and orientation statistics. In both cases, we employed a workflow in which geologic expertise interacts with statistical protocol to motivate geologically relevant statistical tests (Fig. 1). In this framework, statistics connects the collected dataset to the geologic system through quantitative measures of uncertainty. We find significant utility in adopting such a workflow, particularly for datasets that are small and disperse. The statistical approach allowed us to interpret subtle differences in domains as real through hypothesis testing.

Statistical tools are critical to the future of structural geology. As structural geology datasets become available in open source databases, these statistical tools will be increasingly important. When combining datasets collected by different geologists over the same geographic extent, these tools provide a way to test whether combining datasets is permissible. When examining the same type of geologic feature at thousands of field locations worldwide, these tools provide a way to quantitatively compare geometries.

Acknowledgements

This work was supported by funding from the National Science Foundation EAR 1639748 to B. Tikoff and J. Newman. Thank you to V. Chatzaras, N. Garibaldi, R. Williams, A. Jones, C. Bate and the rest of the Structure and Tectonics group at UW-Madison for insightful manuscript comments.

References

Allmendinger, R. W. (2017), Stereonet 10.0.0 [Computer software]. Retrieved from http://www.geo.cornell.edu/geology/faculty/RWA/programs/stereonet.html

Armstrong, R. L., W. H. Taubeneck, and P. O. Hales (1977), Rb-Sr and K-Ar geochronometry of Mesozoic granitic rocks and their Sr isotopic composition, Oregon, Washington, and Idaho, *Geological Society of America Bulletin*, *88*(3), 397–411. https://doi.org/10.1130/0016-7606(1977)88<397:RAKGOM>2.0.CO;2

Benford, B., J. Crowley, M. Schmitz, C. J. Northrup, and B. Tikoff (2010), Mesozoic magmatism and deformation in the northern Owyhee Mountains, Idaho: Implications for along-zone variations for the western Idaho shear zone, *Lithosphere*, *2*(2), 93–118. https://doi.org/10.1130/L76.1

Braudy, N., R. M. Gaschnig, D. Wilford, J. D. Vervoort, C. L. Nelson, C. Davidson, M. J. Kahn, and B. Tikoff (2017), Timing and deformation conditions of the western Idaho shear zone, West Mountain, west-central Idaho, *Lithosphere*, *9*(2), 157–183. https://doi.org/10.1130/L519.1

Davidson, G. F. (1990), Cretaceous tectonic history along the Salmon River suture zone near Orofino, Idaho: Metamorphic structural and $^{40}Ar/^{39}Ar$ thermochronologic constraints (M.Sc. thesis). Location: Oregon State University.

Davis, J. C., and R. J. Sampson (1986), *Statistics and data analysis in geology*, Wiley New York et al.

Davis, J. R., and S. J. Titus (2017), Modern methods of analysis for three-dimensional orientational data, *Journal of Structural Geology*, *96*, 65–89. https://doi.org/10.1016/j.jsg.2017.01.002

Ducharme, G. R. (1985), Bootstrap confidence cones for directional data, *Biometrika*, *72*(3), 637–645. https://doi.org/10.1093/biomet/72.3.637

Feynman, R. P. (1974), Cargo cult science, *Engineering and Science*, *37*(7), 10–13.

Fisher, N. I., and P. Hall (1989), Bootstrap confidence regions for directional data, *Journal of the American Statistical Association*, *84*(408), 996–1002. https://doi.org/10.1080/01621459.1989.10478864

Fleck, R. J., and R. E. Criss (1985), Strontium and oxygen isotopic variations in Mesozoic and Tertiary plutons of central Idaho, *Contributions to Mineralogy and Petrology*, *90*(2–3), 291–308. https://doi.org/10.1007/BF00378269

Fleck, R. J., and R. E. Criss (2004), Location, age, and tectonic significance of the Western Idaho Suture Zone (WISZ). *Report: USGS Numbered Series,* 2004-1039, p. 48.

Giorgis, S., W. McClelland, A. Fayon, B. S. Singer, and B. Tikoff (2008), Timing of deformation and exhumation in the western Idaho shear zone, McCall, Idaho, *Geological Society of America Bulletin*, *120*(9–10), 1119–1133. https://doi.org/10.1130/B26291.1

Giorgis, S., Z. D. Michels, L. Dair, N. Braudy, and B. Tikoff (2017), Kinematic and vorticity analyses of the western Idaho shear zone, USA, *Lithosphere*, *9*(2), *223–234*. https://doi.org/10.1130/L518.1

Jupp, P. E., and K. V Mardia (1989), A unified view of the theory of directional statistics, 1975-1988, *International Statistical Review/Revue Internationale de Statistique*, *57*(3), 261–294. https://doi.org/10.2307/1403799

Lewis, R. S., K. L. Schmidt, R. M. Gaschnig, T. A. LaMaskin, K. Lund, K. D. Gray, B. Tikoff, T. Stetson-Lee, and N. Moore (2014), Hells Canyon to the Bitterroot front: A transect from the accretionary margin eastward across the Idaho batholith, *Geological Society of America Field Guides*, *37*, 1–50.

Mahalanobis, P. C. (1936), On the generalized distance in statistics, *Proceedings of the National Institute of Sciences of India, 1936*, 49–55.

Manduca, C. A., L. T. Silver, and H. P. Taylor (1992), $^{87}Sr/^{86}Sr$ and $^{18}O/^{16}O$ isotopic systematics and geochemistry of granitoid plutons across a steeply-dipping boundary between contrasting lithospheric blocks in western Idaho, *Contributions to Mineralogy and Petrology*, *109*(3), 355–372. https://doi.org/10.1007/BF00283324

Mardia, K. V, and P. E. Jupp (2000), *Directional statistics*, John Wiley & Sons.

Marrett, R., and R. W. Allmendinger (1990), Kinematic analysis of fault-slip data, *Journal of structural geology*, *12*(8), 973–986. https://doi.org/10.1016/0191-8141(90)90093-E

McClelland, W. C., and J. S. Oldow (2004), Displacement transfer between thick-and thin-skinned décollement systems in the central North American Cordillera, *Geological Society, London, Special Publications*, *227*(1), 177–195. https://doi.org/10.1144/GSL.SP.2004.227.01.10

Pollack, H. N. (2003), *Uncertain science... uncertain world*, Cambridge University Press.

Rember, W. C., & Bennett, E. H. (1979). Geologic map of the Idaho Falls quadrangle. *Idaho, Idaho Bureau of Mines and Geology Geologic Map Series, Idaho Falls*, *2*.

Schmidt, K. L., R. S. Lewis, J. D. Vervoort, T. A. Stetson-Lee, Z. D. Michels, and B. Tikoff (2017), Tectonic evolution of the Syringa embayment in the central North American Cordilleran accretionary boundary, *Lithosphere*, *9*(2), 184–204. https://doi.org/10.1130/L545.1

Stetson-Lee, T. A. (2015), Using Kinematics and Orientational Statistics to Interpret Deformational Events: Separating the Ahsahka and Dent Shear Zones Near Orofino, ID (M.Sc. thesis). Location: University of Wisconsin--Madison.

Tsagris, M., and Athineou, G. (2016). Directional: Directional Statistics. R package version 1.8. http://CRAN.R-project.org/package=Directional

Vollmer, F. W. (2017), Software for the quantification, error analysis, and visualization of strain and fold geometry in undergraduate field and structural geology laboratory experiences, in *Geological Society of America Abstracts with Programs*, *49*(2). https://doi.org/10.1130/abs/2017NE-291027

Wellner, J. A. (1979), Permutation tests for directional data, *The Annals of Statistics*, *7*(5), 929–943. http://www.jstor.org/stable/2958664

Yonkee, W. A., and A. B. Weil (2015), Tectonic evolution of the Sevier and Laramide belts within the North American Cordillera orogenic system, *Earth-Science Reviews*, *150*, 531–593. https://doi.org/10.1016/j.earscirev.2015.08.001

Figure Captions

Figure 1. Schematic diagram of a structural geology workflow that takes advantage of statistical tools to aid interpretations of the geologic system. The grey box surrounds the statistical component of the workflow, and is a simplification of the statistical flowchart from Davis and Titus (2017). Railroad ties on arrows indicate portions of the workflow used in the examples in this paper. The structural geologist begins with an incomplete representation of the geologic system (the dataset). After visualizing the data, two simultaneous processes begin—the generation of geologic hypotheses/associated predictive models and a statistical protocol that should be done on any dataset. The statistical protocol interacts when testing the geologic hypotheses either through regressions or statistical hypothesis testing. Importantly, all interpretations of the geologic system run through the grey statistical box.

Figure 2. Simplified geologic map and overview of data from the West Mountain, ID area of the Late Cretaceous western Idaho shear zone published in Braudy et al. (2017). **A)** Geologic units of the western Idaho shear zone (Red—Muir Creek orthogneiss, Purple—Sage Hen orthogneiss, Magenta—Payette River Tonalite). The Muir Creek orthogneiss was the focus of the structural study in Braudy et al. (2017). **B)** Geographic locations and symbols of foliation-lineation data (left) and foliation-only data (right). There are 148 foliation-only measurements and 129 foliation-lineation pairs. **C)** Equal area nets with data for foliation-only (above) and foliation-lineation datasets (below), color-coded by the geographic domains used by Braudy et al. (2017) (Red—northern, Green—central, Blue—southern). Map modified from Braudy et al. (2017).

Figure 3. Summary of the statistical analyses for the West Mountain field fabrics dataset. **A)** An analysis of the claim from Braudy et al. (2017) that there is a 20° rotation between the northern and southern domains: Top, a lower hemisphere equal area projection (with zoomed-in cutout) with the 95% confidence regions for the mean of foliation-only data in each of the three domains (Red—northern, Green—central, Blue—southern) as determined from bootstrapping; Middle, a histogram of angular distances between bootstrap iterations of the northern and southern domains; Bottom, a visualization of the rotation computed from the bootstrapped angular distance and corresponding rotation axes. **B)** A series of two-sample hypothesis tests plotted on equal volume plots (with zoomed-in cutouts). Both bootstrapping and 95% confidence ellipsoids as well as Markov chain Monte Carlo (MCMC) mean probability clouds and their 95% credible ellipsoids are used to compare each pair of domains (Black—northern, Orange—central, Blue—southern). **C)** A comparison of 95% confidence ellipses from bootstrapping foliations. Foliations from foliation-lineation data are compared with those from foliation-only data within each domain: Colors are the same as in (A).

Figure 4. Simplified geologic map of the Orofino area, with the foliation-lineation dataset superimposed. Exposure of sheared Late Cretaceous basement below the Miocene Columbia River basalts is limited to the shoreline of Dvorshak reservoir. An interpretation of the boundary between the Woodrat Mountain and Ahsahka shear zones is shown. Modified from Rember and Bennett (1979).

Figure 5. Summary of statistical analysis for the Orofino, ID area foliation-lineation dataset. **A)** Two different plots of the foliation-lineation data colored by kilometers north: Left, an equal-area plot with lineations (squares) and foliation poles (circles), each with 2σ, 6σ, 10σ, 14σ, and 18σ Kamb contours; Right, an equal volume plot after Davis and Titus (2017) with translucent 2σ Kamb contours. Each point in the equal volume plot is a foliation-lineation pair represented as a rotation from a reference plane-line pair. Note that there are four copies of the dataset due to four-fold symmetry of such data (See Davis and Titus (2017) for more information). **B)** A series of 18 geodesic regressions testing geographic variation along specific azimuths. Each solid dot is a regression with a corresponding *p*-value (open circle). **C)** The geologic map from Figure 5 superimposed with the domains used in this statistical analysis. **D)** A series of two-sample hypothesis tests plotted on equal volume plots (with zoomed-in cutouts). MCMC mean probability clouds and their 95% credible regions as well as bootstrapped mean clouds and their 95% confidence region are used to compare each pair of domains (Black—domain 1, Orange—domain 2, Blue—domain 3). **E)** A lower-hemisphere, equal-area projection showing the results of the MCMC analysis. Colors are the same as for (D).

Table 1. The Fréchet mean strike, dip, and rake for the three domains in the Ahsahka segment of the western Idaho shear zone. Strike/dip/Rake are in right hand rule.

APPENDIX 1: Geologic Background

Both datasets examined in this paper come from the western Idaho–Ahsahka shear zone system, Idaho, which forms a steep and abrupt boundary between the accreted terranes (consisting of collided magmatic arcs) and the cratonic edge of the North American Cordillera. (e.g., Armstrong et al., 1977; Fleck and Criss, 1985; Manduca et al., 1992; Fleck and Criss, 2004; Braudy et al., 2017) and recent seismic data (Stanciu et al., 2016; Davenport et al., 2017). Schmidt et al. (2016) noted that the timing of deformation is nearly identical within the western Idaho and Ahsahka shear zones, and they are in structural continuity.

The first dataset comes from the western Idaho shear zone portion of the boundary, near West Mountain, ID. In this segment, sub-vertical fabric is oriented ~020 in southwestern Idaho (Benford et al., 2010) and 000 in central Idaho (Manduca et al., 1993; Giorgis et al., 2008). The western Idaho shear zone is ~5 km wide and is characterized by highly deformed orthogneisses. Deformation is interpreted as dextral transpressional, with a vertical vorticity axis parallel to the lineation orientation (e.g., Giorgis et al., 2008; Michels et al., 2015) and an estimated angle of oblique convergence of 45-60° (Giorgis et al., 2016). At West Mountain, metamorphosed sedimentary wall rocks involved in the shear zone record pressures of 4.5 kbar and temperatures of ~730°C (Braudy et al., 2017). Deformation occurred in the middle Cretaceous, constrained regionally to have occurred between 104-90 Ma with peak metamorphism at 100-97 Ma (Braudy et al., 2017). Of particular relevance to this study is that Braudy et al. (2017) noted a change in foliation orientation in the West Mountain segment, which they interpreted as reflecting a primary along-strike variation in the western Idaho shear zone.

The second dataset comes from the Ahsahka shear zone portion of the boundary. The Ahsahka shear zone occurs in a 90° change in orientation in the fabric of the zone near Orofino, ID (e.g., Lewis et al., 2014). The NS orientation of the WISZ rotates to an EW orientation in the Ahsahka shear zone.

The exact nature of deformation around Orofino, ID, is confusing because multiple shear zones are located in the area and abundant flows of the Miocene Columbia River basalt group cover the region. Both Strayer et al. (1989) and Davidson (1990) recognized zones of ductilely deformed rocks and determined top-to-the-south/southwest shear sense indicators. These fabrics typically occur in mylonitic gneisses that display moderately north-northeast dipping foliation and steeply pitching lineation. Schmidt et al. (2016) designated these fabrics to be part of the Ahsahka shear zone, which is distinct from the older Woodrat Mountain shear zone (Lewis et al., 2014) to the northeast. Deformation in the Ahsahka shear zone occurred between 116-89 Ma (Schmidt et al., 2016), with dominantly reverse-sense shear and a horizontal vorticity vector perpendicular to the lineation direction (Giorgis et al., 2017). Critical for this study is that there are multiple shear zones in the Orofino area, including the older Woodrat shear zone (McClelland and Oldow, 2004, 2007; Lund et al., 2007).

APPENDIX 2: **A primer on the inference methods used in this paper**

Regressions

Geographic gradients are commonly observed in structural geology. Two common examples are strain gradients or change in orientation due to gentle regional folding. The observation of these gradients plays a major role in how we interpret data. Regressions are designed to detect and quantify the strength of these gradients.

In elementary statistics, the simplest regression takes the form of a best fit line, which minimizes the square of the distances between the data points and that line. The analogously simple regression for orientation data is a *geodesic regression* (Davis and Titus, 2017). In our statistical analysis, a best-fit geodesic curve describes the steady-state relationship between orientation of fabric and an azimuthal direction. For example, if fabric rotates about a specific axis 5° every 100 meters to the northwest (a 45° azimuth), then a geodesic curve would accurately describe the geographic gradient. The R^2 of the regression provides a description of how closely the data fit the geodesic function. An R^2 of 1 represents a perfect fit, and an R^2 of 0 represents no fit.

3.2 Inference methods

Inference methods use the dataset to test hypotheses about the population. In this paper, we focus on two approaches to inference. When dealing with directional data like foliations, we use *bootstrapping* in addition to three more standard directional statistical tests. When dealing with orientation data like foliation-lineation pairs, we use both *bootstrapping* and a *Markov chain Monte Carlo* simulation.

Bootstrapping is non-parametric, meaning that it does not depend on an assumption of distribution (Efron and Tibshrani, 1994). Simply put, bootstrapping is repeated resampling with replacement. For example, if there are 20 foliation measurements, then each iteration of the bootstrapping routine selects 20 measurements from the dataset, but some of the original data are picked more than once and others not at all. We use 10,000 iterations. The mean of each of these perturbed iterations is recorded. When plotted, the density of the cloud of bootstrapped means reflects the sampling distribution of the mean, which is the concept that underlies confidence regions. The 95% confidence region around the mean of the dataset describes the uncertainty in the sample mean as an estimate of the population mean. It is computed by using principal component analysis and the Mahalanobis distance (Mahalanobis, 1936) to fit an ellipsoid that contains 95% of the bootstrapped means.

The *Markov chain Monte Carlo* simulation takes an entirely different, probabilistic approach to quantifying uncertainty about the population mean. The method is a Bayesian approach; instead of asking how probable the dataset is given the statistical model (e.g. how well the data fit a normal distribution with a specific mean and variance), the Bayesian approach asks the opposite question: How probable a statistical model is given the dataset? The question posed in this way is called the *posterior probability*. The *Markov chain Monte Carlo* algorithm produces a set of iteratively perturbed statistical models which converge on the statistical model that best fits the data. In each step of the algorithm, the parameters of the statistical model from the previous step are modified slightly, and the posterior probability is calculated. If the posterior probability is better than it was for the previous step, then the new parameters are kept. Otherwise, the new step has a chance of being overwritten by the previous step. The value of the mean in each of these perturbed statistical models creates a list of means similar to bootsrapping, where the density of the means when plotted reflects the relative probability that a point represents the true population mean. A region containing 95% of the means is computed using Mahalanobis distance as above. This region is called a 95% *credible* region in recognition of its Bayesian origin. In practice we use credible regions and confidence regions similarly.

We use both bootstrapping and MCMC simulation because they each have advantages and weaknesses. In general, for small sample sizes ($n < 30$), MCMC returns credible regions with accurate size, but the regions tend to be unrealistically isotropic. By contrast, bootstrapping returns more realistically proportioned confidence regions, but the size of the region is consistently underestimated.

For foliations (directional data), we also employ three two-sample tests from the literature as a check on the bootstrap methodology. The Wellner two-sample test (Wellner, 1979) computes a *T*-statistic that provides a measure of how different the two samples are. The samples are then repeatedly scrambled with one another, and the *T*-statistic is calculated for each iteration. The percentage of *T*-statistics that are greater than the *T*-statistic of the original datasets (instances where scrambled datasets were more dissimilar than the original) serves as a *p*-value. The other two-sample tests are versions of the Watson two-sample test, in which an *F* statistic is computed and plotted on an *F* distribution for comparison with a $p = 0.05$ threshold value. The two Watson two-sample tests make different assumptions to approximate the *F* statistic; one assumes that the dataset is large, the other assumes that the data are tightly concentrated (Mardia and Jupp, 2000).

3.3 The null hypothesis

The inference methods described above are used to test *null hypotheses.* A null hypothesis is a statement that can be rejected (or fails to be rejected) based on inference. For example, a null hypothesis for a foliation dataset might be "the mean of the population is a foliation that strikes 077 and dips 35." If 077/35 is outside of the 95% confidence bootstrap region, then we can reject the null hypothesis at 95% confidence. If, however, 077/35 falls within the 95% confidence region, we cannot reject the null hypothesis. Note that in the case that we fail to reject the null hypothesis, we *do not accept* it. Failure to reject the null hypothesis is an inconclusive result.

For the most part, this paper uses confidence/credible regions for testing the null hypothesis, because both bootstrapping and Markov chain Monte Carlo approaches result in confidence/credible regions, and null hypotheses are rejected or not based on whether or not the hypothesized value lies within these regions. We reject the null hypothesis when the confidence is

95% or more. The Wellner and Watson two-sample tests report *p*-values. We reject the null hypothesis when $p < 0.05$.

APPENDIX 3: West Mountain data

Appendices 3, 4, 5, and 6 are data files and R scripts used for the statistical analysis. Follow the directions below to access and run the scripts and data files in their proper format.

1. Go to http://www.joshuadavis.us/software/ and download the *geologyGeometry* R library as well as the zip file containing:
    a. *JSG_statsFunctions.r*
    b. *JSG_statisticalAnalsysis_WestMountain.r*
    c. *JSG_statisticalAnalysis_Orofino.r*
    d. *Fols_WestMt.csv*
    e. *Follins_WestMt.csv*
    f. *Follins_Ahs.csv*
2. Follow the instructions in section 2 of the *readme.pdf* within the *geologyGeometry* folder. The installation of R, R-studio, and required R packages is described.
3. Move the *geologyGeometry* folder out of the Downloads folder to a location of your choosing and save the three R scripts (*.r*) into the *geologyGeometry* folder.
4. Save the three datasets (*.csv*) in the *data* folder within *geologyGeometry*
5. At the top of each script, there is a line setwd("~/Desktop/20170620geologyGeometry") Change the text within the quote to match the path on your computer of the *geologyGeometry* folder.

West Mountain Foliations

location	easting	northing	strike	dip
10B352	562899	4915537	0	81
12b6	564099	4925419	0	76
12b9	565249	4925632	0	84
12b9	565249	4925632	0	80
12b10	565508	4925751	0	87
12b13	565492	4925823	0	80
12b19	563193	4920861	0	80
12b28	563550	4912053	0	75
12b73	564367	4925408	0	82
12b74	564609	4925466	0	83
12b9	565249	4925632	1	84
12b6	564099	4925419	2	80
12b9	565249	4925632	2	88
12b73	564367	4925408	2	88
12b74	564609	4925466	2	80
12b6	564099	4925419	3	75
12b6	564099	4925419	3	80
12b7	564258	4925370	3	83
12b9	565249	4925632	3	87
10B347	561990	4916474	4	76
12b63	564555	4925819	4	82
12b73	564367	4925408	4	79
186	561654	4911123	5	81
12b7	564258	4925370	5	73
12b7	564258	4925370	5	84
12b9	565249	4925632	5	81
12b9	565249	4925632	5	82
12b9	565249	4925632	5	90
12b73	564367	4925408	5	82
12b6	564099	4925419	6	79
12b6	564099	4925419	6	77
12b6	564099	4925419	6	81
12b7	564258	4925370	6	81
12b7	564258	4925370	6	85
12b9	565249	4925632	6	85
10b18	563252	4912347	7	77

10B340	562668	4916013	8	81
184	562273	4911342	8	68
12b6	564099	4925419	8	77
12b6	564099	4925419	8	81
12b6	564099	4925419	8	80
12b18	563126	4920162	8	84
12b74	564609	4925466	8	80
12b6	564099	4925419	9	75
12b7	564258	4925370	9	80
12b7	564258	4925370	9	80
12b49	561431	4922977	9	76
12b52	563344	4924630	9	78
10B260	564573	4916371	10	75
10B298	563985	4917688	10	68
10B317	562561	4919112	10	76
342	563437	4922678	10	85
12b6	564099	4925419	10	85
12b6	564099	4925419	10	72
12b9	565249	4925632	10	89
12b54	560198	4915017	10	84
12b73	564367	4925408	10	80
12b15	566528	4925343	11	81
10B294	564714	4918101	12	71
10B350	562181	4914854	12	70
12b9	565249	4925632	12	90
257	563383	4918160	13	80
192	563018	4911014	13	80
281	562606	4922570	13	74
12b74	564609	4925466	13	85
12b74	564609	4925466	13	84
12b33	563237	4916238	14	65
10B252	563182	4921127	15	76
10B335	562734	4910230	15	75
353	563856	4924651	15	70
158	560823	4920948	15	80
12b1	564064	4919837	15	75
12b22	564910	4922012	15	83
256	564196	4917699	16	76
12b41	561281	4913879	16	80
12b74	564609	4925466	16	81

10B278	565277	4920296	18	78
307	563652	4920458	18	75
311	563819	4921753	18	86
270	563203	4914508	18	80
265	562183	4914853	18	72
12b29	563356	4911550	18	82
10B343	562093	4917361	19	69
226	564325	4916243	19	68
343	564066	4922584	20	74
347	564202	4923510	20	85
232	563145	4918523	20	76
10b339	562916	4916258	21	77
317	561254	4915050	21	78
12b35	562083	4916519	21	73
12b42	562278	4914361	21	85
200	562424	4910135	22	78
215	561358	4911562	22	82
222	562320	4911899	22	84
12b92	559931	4914940	22	80
252	564360	4919255	23	83
267	563278	4914963	23	84
219	562149	4912372	23	76
262	561058	4915442	23	70
12b34	562545	4916573	23	80
12b37	561122	4915981	23	83
12b20	563579	4921650	24	82
12b53	559727	4915063	24	84
213	562414	4910680	25	80
217	561601	4911802	25	75
12b21	564148	4922080	25	84
291	561226	4921280	26	78
319	563260	4923531	26	88
12b31	562199	4911178	26	78
12b40	560441	4914577	26	78
12b81	561353	4911377	26	84
244	562086	4916520	27	74
183	562385	4911331	28	70
246	561112	4916706	28	84
280	562665	4920857	28	74
284	562343	4920861	28	82

306	563613	4920114	30	79
242	563007	4916431	30	80
338	560845	4921119	30	76
12b32	561675	4911115	30	85
254	564448	4918489	31	82
261	560811	4915223	31	70
228	564391	4916670	32	72
230	564096	4915684	32	82
313	564469	4921361	33	78
205	563502	4913000	33	82
272	562810	4914633	34	85
314	564291	4920918	35	84
334	564259	4925347	35	80
12b30	562269	4911430	35	84
214	561405	4911320	36	80
247	561082	4916561	36	76
250	563828	4919173	38	85
255	564141	4918083	43	80
354	564722	4924615	45	83
12b38	560845	4916189	190	20
10B305	562397	4911536	233	25
241	561125	4917141	240	74
10B349	562041	4915544	350	78
10B297	564330	4917548	353	74
10B321	561292	4917601	353	75
10B286	564402	4919040	354	66
10B346	561624	4916926	354	66
12b73	564367	4925408	354	79
10B312	562092	4913318	358	78
290	561247	4921925	358	81
12b7	564258	4925370	358	85
12b6	564099	4925419	359	85

West Mountain Foliation-lineation pairs

location	easting	northing	strike	dip	rake
10NB251	563187	4920843	0	75	84
10NB296	564303	4917183	1	74	84
10NB288	564939	4919194	1	78	109
10NB334	562780	4910922	2	83	106

10nb15	563700	4914745	2	79	70
10NB353	563146	4916000	2	70	86
10NB259	564014	4916127	2	80	94
10NB284	564090	4918880	2	76	68
12nb58	565161	4925777	3	90	90
10NB351	562545	4914749	4	70	83
10NB295	564581	4916838	4	74	94
10NB345	561750	4917349	4	79	81
12nb74	564609	4925466	4	84	88
12nb14	565854	4925645	4	82	87
10NB318	562379	4918476	5	87	85
10NB274	563763	4919693	5	68	49
292	561008	4921378	5	68	88
12nb8	565187	4925595	5	84	90
10NB303	563145	4911642	6	74	88
10NB320	561771	4918228	6	83	106
10NB282	563709	4918710	6	74	89
10nb256	562933	4920045	6	82	76
10NB254	563759	4921942	6	76	107
10NB255	564296	4922089	6	72	81
10NB299	563764	4917245	7	66	77
316	559640	4915004	8	77	79
225	563998	4915962	8	74	83
233	562818	4919310	8	76	76
10NB270	562907	4920261	8	67	60
12nb73	564367	4925408	8	83	90
12nb9	565249	4925632	8	79	92
10NB304	562493	4911386	9	81	96
10NB283	564069	4918528	9	55	64
10NB272	562508	4920909	9	75	104
10nb21	560176	4914984	10	74	98
10NB341	562284	4916038	10	55	80
10NB322	561574	4917466	10	53	75
10NB275	564084	4919796	10	74	86
10NB276	564369	4920044	10	71	94
12nb48	561006	4921364	10	65	100
337	564778	4925527	10	81	94
237	561946	4918274	11	86	72
10NB285	564201	4919037	11	78	71
348	563819	4923745	11	85	77

10NB271	562765	4920729	13	70	94
10NB302	563437	4911950	14	67	97
10NB323	561806	4917596	14	75	86
10NB277	564575	4920338	14	69	87
10NB306	561585	4911310	15	75	101
10NB315	562219	4912076	15	77	102
10NB314	562119	4912641	15	80	87
10NB313	562060	4912935	15	76	53
10NB348	562017	4916269	15	67	68
287	560766	4919763	15	83	99
10NB336	562484	4909840	16	65	86
10NB337	562862	4910336	16	84	106
253	564162	4918708	16	82	78
10NB287	564619	4919104	16	83	92
345	564813	4922999	16	86	120
12nb6	564099	4925419	16	80	82
10NB309	561398	4912922	17	75	96
10NB300	563630	4917000	17	76	98
10NB269	562537	4920037	17	76	95
180	563219	4911445	18	78	60
206	563355	4913162	18	76	98
271	562921	4914587	18	84	72
10nb20	559716	4915012	18	80	74
227	564460	4916503	18	70	70
243	562536	4916523	18	85	103
276	562903	4920219	18	78	86
283	562139	4921509	18	80	96
341	563129	4922574	18	70	80
293	561411	4922912	18	84	72
304	561572	4923631	18	75	72
204	563447	4912616	19	77	98
10NB311	562016	4913416	19	80	86
278	562773	4920762	19	70	83
10NB253	563408	4921898	19	74	72
245	561466	4916675	20	65	96
10NB250	563281	4920358	20	74	83
336	561302	4925354	20	82	88
203	563465	4912283	21	78	102
10NB308	561391	4912787	21	77	98
181	562848	4911146	22	82	102

10nb19	562197	4911926	22	79	90
273	562499	4919385	22	84	88
275	562606	4920092	22	79	98
344	564770	4922618	22	85	88
294	561498	4923408	23	84	86
349	563386	4923719	23	83	76
224	563924	4915661	24	75	82
251	564046	4919273	24	82	73
310	563570	4921601	24	81	96
191	563032	4911027	25	80	92
207	563206	4913291	25	85	105
12nb36	561614	4916371	25	85	80
236	562544	4918749	25	82	65
335	564258	4925347	25	68	90
216	561404	4911631	27	71	102
258	560911	4916388	28	76	84
249	561643	4916771	28	71	100
309	563594	4921409	28	84	98
269	563258	4914518	29	80	83
218	561763	4912260	32	78	104
268	563536	4914451	32	82	100
202	563512	4912056	33	85	84
318	561510	4912827	36	77	73
185	562022	4911292	38	55	98
229	564252	4916791	38	80	75
10NB301	563768	4916433	166	78	116
10NB333	562945	4911224	169	85	105
10NB319	562177	4918387	171	83	87
10NB376	562668	4919460	175	79	90
10NB257	562733	4919640	181	72	80
10NB268	562491	4919388	197	88	98
10nb17	563439	4913213	330	52	102
10NB325	562225	4917845	350	86	105
10NB331	563148	4910998	352	75	79
315	560210	4915033	353	85	83
10NB327	562898	4918849	353	79	87
10nb16	562864	4915344	354	82	98
10NB310	561791	4913562	355	75	101
10NB342	562374	4916633	355	80	94
10NB316	563126	4919207	355	62	104

12nb27	565758	4922420	355	75	93
10NB326	562682	4918413	356	75	83
10NB273	563192	4921417	356	75	95
10NB344	562264	4917415	357	79	92
10NB324	561897	4917860	357	61	100

APPENDIX 4: West Mountain statistical analysis

```
### PRELIMINARY WORK ####

# Set the working directory.
setwd("~/Desktop/20170620geologyGeometry")
# Load the necessary R libraries.
source("library/all.R")

# Load any custom functions called in this file####

#one Regression does a regression based on an azimuth (e.g. how well
does northeasting (045) explain variation in my data?) This kernal
geodetic regression is analogous to a best-fit line, and thus describes
a steady variation, not a sudden change in orientation.
oneRegression <-
  function(follins,
           domain,
           pValuePerms,
           directionality) {
    regression <-
      oriGeodesicRegression(
        cos((directionality) * pi / 180) * follins$northing[domain] +
sin((directionality) *

pi / 180) * follins$easting[domain],
        follins$rotation[domain],
        oriLineInPlaneGroup,
        numSteps = 10000
      )

    if (pValuePerms > 0) {
      RSquareds <-
        oriGeodesicRegressionPermutations(
          cos((directionality) * pi / 180) * follins$northing[domain] +
sin((directionality) *

pi / 180) * follins$easting[domain],
          follins$rotation[domain],
          numPerms = pValuePerms,
          group = oriLineInPlaneGroup
        )
      sum(RSquareds > regression$rSquared)
      p <- sum(RSquareds > regression$rSquared) / length(RSquareds)
    } else {
```

```
      p <- 0
    }
    regressionStats <- zeros(1, 5)
    regressionStats[1, 1] = directionality
    regressionStats[1, 2] = regression$error
    regressionStats[1, 3] = regression$minEigenvalue
    regressionStats[1, 4] = regression$rSquared
    regressionStats[1, 5] = p
    names(regressionStats) = c("azimuth", "error", "minEigenValue",
"R^2", "P")
    regression$stats <- regressionStats
    return(regression)
  }

#regressionSweep does a series of directional regressions from 0-180°,
given an increment, and also calculates the pValue for each regression.
BEWARE: doing this with p-values can take days-weeks-or-months of
computing time. I suggest first running the function with degreeIncrement
= 10 and pValuePerms=0. This may still take a couple hours, but will
give you a first order picture of whether it is interesting to proceed.
regressionSweep <-
  function(follins,
           degreeIncrement,
           domain,
           pValuePerms) {
    v <- rbind(1)
    intervals <- 180 / degreeIncrement
    for (x in 2:intervals) {
      vTemp <- x
      v <- rbind(v, vTemp)
    }
    v <- as.matrix(as.vector(v))
    # Geodesic regression of pole vs. northing in domain 4. Check that
error is 0 and minEigenvalue is positive.
    regressions <- list()
    regressionStats <- zeros(nrow(v), 5)
    for (i in 1:nrow(v)) {
      regressionTemp <-
        oriGeodesicRegression(
          cos((v[i, 1] - 1) * degreeIncrement * pi / 180) *
follins$northing[domain] + sin((v[i, 1] -

1) * degreeIncrement * pi / 180) * follins$easting[domain],
          follins$rotation[domain],
          oriLineInPlaneGroup,
          numSteps = 10000
        )
      regressions[[i]] <- regressionTemp
      regressionStats[i, 1] = (i - 1) * degreeIncrement
      regressionStats[i, 2] = regressionTemp$error
      regressionStats[i, 3] = regressionTemp$minEigenvalue
      regressionStats[i, 4] = regressionTemp$rSquared
      if (pValuePerms > 0) {
```

```
        RSquareds <-
          oriGeodesicRegressionPermutations(
            cos((v[i, 1] - 1) * degreeIncrement * pi / 180) * follins$northing[domain] + sin((v[i, 1] -1) * degreeIncrement * pi / 180) * follins$easting[domain],
            follins$rotation[domain],
            numPerms = pValuePerms,
            group = oriLineInPlaneGroup)
        length(RSquareds)
        sum(RSquareds > regressionTemp$rSquared)
        p <-
          sum(RSquareds > regressionTemp$rSquared) / length(RSquareds)
      }
      else {
        p <- "nan"
      }

      regressionStats[i, 5] = p
    }
    regressions[[i + 1]] <- regressionStats
    return(regressions)
  }

#take two bootstrapped mean clouds and the number of data points in each, and calculate a distribution of angular differences. Probably not the most statistically robust.
distHist <- function(bootsOne, bootsTwo, numPerms, group) {
  distances <- list()
  boots1 <- bootsOne$bootstraps
  boots2 <- bootsTwo$bootstraps
  for (i in 1:numPerms) {
    a <- round(runif(1, 1, length(boots1)), 0)
    b <- round(runif(1, 1, length(boots2)), 0)
    distances[[i]] <- oriDistance(boots1[[a]], boots2[[b]], group)
  }

  return(distances)
}

#Using one two-sample bootstrapped mean difference cloud, calculate the distribution of angular differences. Probably not the most statistically robust.
diffDistHist <- function(bootsDiff, numPerms, group) {
  boots <- bootsDiff$bootstraps
  distances <-
    lapply(boots, function(s)
      oriDistance(s, diag(3), group) * 180 / pi)
  return(distances)
}
```

END SCRIPT

```r
### PRELIMINARY WORK ####

# Set the working directory.
setwd("~/Desktop/20170620geologyGeometry")
# Load the necessary R libraries.
source("library/all.R")
# Markov chain Monte Carlo and Kamb contouring in equal volume plots
require C compiler. Skip MCMC and equal volume Kamb lines if you do not
wish to install C. Load the necessary library
source("libraryC/all.R")
# Load some custom functions.
source("JSG_statsFunctions.r")

#===========================LOAD THE DATA========================================

# 1) Foliation only
Fols <- geoDataFromFile("data/Fols_WestMt.csv")

# Plot foliation locations in map view
plot(Fols$easting, Fols$northing, xlab = "Easting (meters)", ylab =
"Northing (meters)")

# Check how many measurments there are
nrow(Fols)

#————————————————————————————————————————————————————————————

# 2) Foliation-lineation pairs
Follins <- geoDataFromFile("data/Follins_WestMt.csv")

# Plot foliation-lineation locations in map view
plot(
    Follins$easting,
    Follins$northing,
    xlab = "Easting (meters)",
    ylab = "Northing (meters)"
)

# Check how many measurements there are
nrow(Follins)



#===========================PART I: DIRECTIONAL STATISTICS ON FOLIATIONS=========================================
```

```
#==========================DEFINE GEOGRAPHIC DOMAINS FOR FOLIATION-ONLY DATA=================================

#(After Braudy et al., 2017)

# Define the geographic criteria that defines the different domains
Fols_northCrit <- Fols$northing > 4922736
Fols_centerCrit <- Fols$northing < 4922736 & Fols$northing > 4919205
Fols_southCrit <- Fols$northing < 4919205

# Create a new column in the dataframe in which to store the domain information
Fols$domain <- replicate(nrow(Fols), 1)

# Classify the foliation-only dataset by domain
Fols$domain[Fols_northCrit] <- 1
Fols$domain[Fols_centerCrit] <- 2
Fols$domain[Fols_southCrit] <- 3

# Plot the locations of the foliation-only data in map view, each domain a different color.
plot(
    x = Fols$easting,
    y = Fols$northing,
    xlab = "Easting (meters)",
    ylab = "Northing (meters)",
    col = hues(Fols$domain),
    pch = 19
)

# Return the number of datapoints in each domain.
length(Fols$domain[Fols_northCrit])
length(Fols$domain[Fols_centerCrit])
length(Fols$domain[Fols_southCrit])

#==========================DEFINE GEOGRAPHIC DOMAINS FOR FOLIATION-LINEATION DATA========================================

# (After Braudy et al., 2017)

# Define the geographic criteria that defines the different domains
Follins_northCrit <-  Follins$northing > 4922736
Follins_centerCrit <- Follins$northing < 4922736 & Follins$northing > 4919205
Follins_southCrit <-  Follins$northing < 4919205

# Create a new column in the dataframe in which to store the domain information
Follins$domain <- replicate(nrow(Follins), 1)

# Classify the foliation-lineation dataset by domain
Follins$domain[Follins_northCrit] <- 1
Follins$domain[Follins_centerCrit] <- 2
```

```
Follins$domain[Follins_southCrit] <- 3

# Plot the locations of the foliation-lineation data in map view, each
domain a different color.
plot(
    x = Follins$easting,
    y = Follins$northing,
    xlab = "Easting (meters)",
    ylab = "Northing (meters)",
    col = hues(Follins$domain),
    pch = 19
)

# Return the number of datapoints in each domain.
length(Follins$domain[Follins_northCrit])
length(Follins$domain[Follins_centerCrit])
length(Follins$domain[Follins_southCrit])

#==========================FOLIATION-ONLY                    STATISTCAL
TREATMENT=========================================

# 1) Some parametric two-sample tests. The null hypothesis for all tests
is that the two domains being tested come from the same population.

# Three Wellner tests (Wellner, 1979), one for each pair of domains.
Each test is based on 10,000 permutations. This tests not only if the
samples come from the populations with the same mean, but also with the
same dispersion.
lineWellnerInference(Fols$pole[Fols_northCrit],
Fols$pole[Fols_southCrit], 10000)
lineWellnerInference(Fols$pole[Fols_northCrit],
Fols$pole[Fols_centerCrit], 10000)
lineWellnerInference(Fols$pole[Fols_centerCrit],
Fols$pole[Fols_southCrit], 10000)

# Three Watson tests that assume large sample size (Mardia and Jupp,
2000), one for each pair of domains.
lineLargeMultiSampleWatsonInference(list(Fols$pole[Fols_northCrit],
Fols$pole[Fols_southCrit]))
lineLargeMultiSampleWatsonInference(list(Fols$pole[Fols_northCrit],
Fols$pole[Fols_centerCrit]))
lineLargeMultiSampleWatsonInference(list(Fols$pole[Fols_centerCrit],
Fols$pole[Fols_southCrit]))

# Three Watson tests that assume tightly concentrated datasets (Mardia
and Jupp, 2000), one for each pair of domains.
lineConcentratedMultiSampleWatsonInference(list(Fols$pole[Fols_northCr
it],  Fols$pole[Fols_southCrit]))
lineConcentratedMultiSampleWatsonInference(list(Fols$pole[Fols_northCr
it],  Fols$pole[Fols_centerCrit]))
lineConcentratedMultiSampleWatsonInference(list(Fols$pole[Fols_centerC
rit], Fols$pole[Fols_southCrit]))
```

```
#————————————————————————————————————————————————————————————

# 2) Non-parametric bootstrapping

# Perform the bootstrapping routine for each domain. Each bootstrapped
dataset is based on 10,000 iterations.
Fols_northBoots  <-   lineBootstrapInference(Fols$pole[Fols_northCrit],
10000, numPoints = 50)
Fols_centerBoots  <-  lineBootstrapInference(Fols$pole[Fols_centerCrit],
10000, numPoints = 50)
Fols_southBoots  <-   lineBootstrapInference(Fols$pole[Fols_southCrit],
10000, numPoints = 50)

# Plot data for each domain. Northern (red), central (green), southern
(blue)
lineEqualAreaPlotThree(
                       Fols$pole[Fols_northCrit],
                       Fols$pole[Fols_centerCrit],
                       Fols$pole[Fols_southCrit]
)

# Plot the bootstrapped mean clouds for each domain.  Northern (red),
central (green), southern (blue)
lineEqualAreaPlotThree(
    Fols_northBoots$bootstraps,
    Fols_centerBoots$bootstraps,
    Fols_southBoots$bootstraps
)

#  Plot  the  bootstrapped  mean  clouds  with  95%  confidence  ellipses
superimposed.  Northern (red), central (green), southern (blue)
lineEqualAreaPlotThree(
    list(Fols_northBoots$center),
    list(Fols_centerBoots$center),
    list(Fols_southBoots$center),
    curves = list(
        Fols_northBoots$points,
        Fols_centerBoots$points,
        Fols_southBoots$points
    )
)

#————————————————————————————————————————————————————————————

# 3) Compute the rotation between the north and south domains.

# Create empty lists to store data
northSouthDiff <- list()

# Assign variables for the orientations of all bootstrapped means.
northB <- Fols_northBoots$bootstraps
southB <- Fols_southBoots$bootstraps
```

```
# Calculate the smallest possible rotations that bring means from the
northern bootstrap cloud to the southern bootstrap clouds.
count = 1
for (i in 1:100) {
    for (n in i:100) {
        northSouthDiff$rotation[[count]]                              =
rotSmallestRotationFromTwoLines(northB[[i]], southB[[n]])
        if (n < 1000) {
            count = count + 1
        }
    }
}

# Display the raw results of the preceting for loop
northSouthDiff

# Plot these rotations in an equal volume plot
oriEqualVolumePlot(
                   northSouthDiff$rotation,
                   group = oriLineInPlaneGroup,
                   simplePoints = TRUE
)

# Compute the axis and rotation amount from all the rotations
Fols_angleAxis    <-    lapply(northSouthDiff$rotation,    function(s)
rotAxisAngleFromMatrix(s))

# Plot the rotation axes in an equal area plot
lineEqualAreaPlot(
                  lapply(Fols_angleAxis,   function(s)   c(s[1],   s[2],
s[3])),
                  shapes = '.'
)

# Plot a histogram of rotation amount between the northern domain and
the southern domain, in degrees.
hist(
     as.numeric(lapply(Fols_angleAxis, function(s) s[4] * 180 / pi)),
     30,
     xlab = "Angular Distance (degrees)",
     ylab = "Frequency",
     main = "Angular distances, Northern to Southern"
)

# Compute the mean and 1-sigma standard deviation of the rotation amount,
in degrees
mean(sapply(Fols_angleAxis, function(s) s[4] * 180 / pi))
sd(sapply(Fols_angleAxis, function(s) s[4] * 180 / pi))

# Compute the mean trend and plunge in degrees
Fols_axes <- lapply(Fols_angleAxis, function(s) c(s[1], s[2], s[3]))
Fols_meanAxisDeg                                                      <-
geoTrendPlungeDegFromCartesian(lower(lineProjectedMean(Fols_axes)))
```

```
#Print the mean axis of rotation, in trend and plunge
Fols_meanAxisDeg

#==========================FOLIATIONS FROM FOLIATION-LINEATION PAIRS
STATISTICAL TREATMENT========================================

#### FOLIATIONS FROM FOLIATION-LINEATION PAIRS STATISTICAL TREATMENT

# 1) Some parametric two-sample tests.

# Three Wellner tests (Wellner, 1979), one for each pair of domains.
Each test is based on 10,000 permutations.
lineWellnerInference(Follins$pole[Follins_northCrit],
Follins$pole[Follins_southCrit], 10000)
lineWellnerInference(Follins$pole[Follins_northCrit],
Follins$pole[Follins_centerCrit], 10000)
lineWellnerInference(Follins$pole[Follins_centerCrit],
Follins$pole[Follins_southCrit], 10000)

# Three Watson tests that assume large sample size (Mardia and Jupp,
2000), one for each pair of domains.
lineLargeMultiSampleWatsonInference(list(Follins$pole[Follins_northCri
t], Follins$pole[Follins_southCrit]))
lineLargeMultiSampleWatsonInference(list(Follins$pole[Follins_northCri
t], Follins$pole[Follins_centerCrit]))
lineLargeMultiSampleWatsonInference(list(Follins$pole[Follins_centerCr
it], Follins$pole[Follins_southCrit]))

# Three Watson tests that assume tightly concentrated datasets (Mardia
and Jupp, 2000), one for each pair of domains.
lineConcentratedMultiSampleWatsonInference(list(Follins$pole[Follins_n
orthCrit], Follins$pole[Follins_southCrit]))
lineConcentratedMultiSampleWatsonInference(list(Follins$pole[Follins_n
orthCrit], Follins$pole[Follins_centerCrit]))
lineConcentratedMultiSampleWatsonInference(list(Follins$pole[Follins_c
enterCrit], Follins$pole[Follins_southCrit]))

#————————————————————————————————————————————————————————

# 2) Non-parametric bootstrapping

# Perform the bootstrapping routine for each domain. Each bootstrapped
dataset is based on 10,000 iterations.
FollinFols_northBoots <-
lineBootstrapInference(Follins$pole[Follins_northCrit], 10000,
numPoints = 50)
FollinFols_centerBoots <-
lineBootstrapInference(Follins$pole[Follins_centerCrit], 10000,
numPoints = 50)
FollinFols_southBoots <-
lineBootstrapInference(Follins$pole[Follins_southCrit], 10000,
numPoints = 50)
```

```
# Plot data for each domain. Northern (red), central (green), southern
(blue)
lineEqualAreaPlotThree(
                       Follins$pole[Follins_northCrit],
                       Follins$pole[Follins_centerCrit],
                       Follins$pole[Follins_southCrit]
)

# Plot the bootstrapped mean clouds for each domain. Northern (red),
central (green), southern (blue)
lineEqualAreaPlotThree(
                       FollinFols_northBoots$bootstraps,
                       FollinFols_centerBoots$bootstraps,
                       FollinFols_southBoots$bootstraps
)

# Plot the bootstrapped mean clouds with 95% confidence ellipses
superimposed. Northern (red), central (green), southern (blue)
lineEqualAreaPlotThree(
                       list(FollinFols_northBoots$center),
                       list(FollinFols_centerBoots$center),
                       list(FollinFols_southBoots$center),
                       curves = list(
                                     FollinFols_northBoots$points,
                                     FollinFols_centerBoots$points,
                                     FollinFols_southBoots$points
                        )
)å

#————————————————————————————————————————————

# 3) Compute the rotation between the north and south domains.

# Create empty lists to store data
Follins_northSouthDif <- list()

# Assign variables for the orientations of all bootstrapped means.
Follins_northB <- FollinFols_northBoots$bootstraps
Follins_southB <- FollinFols_southBoots$bootstraps

# Calculate the smallest possible rotations that bring means from the
northern bootstrap cloud to the southern bootstrap clouds.
count = 1
for (i in 1:100) {
    for (n in i:100) {
        Follins_northSouthDif$rotation[[count]]                       =
rotSmallestRotationFromTwoLines(Follins_northB[[i]],
Follins_southB[[n]])
        if (n < 1000) {
            count = count + 1
        }
```

```
    }
}

# Display the raw results of the preceting for loop
Follins_northSouthDif

# Plot these rotations in an equal volume plot
oriEqualVolumePlot(
                   Follins_northSouthDif$rotation,
                   group = oriLineInPlaneGroup,
                   simplePoints = TRUE
)

# Compute the axis and rotation amount from all the rotations
Follins_angleAxis <- lapply(Follins_northSouthDif$rotation, function(s)
rotAxisAngleFromMatrix(s))

# Plot the rotation axes in an equal area plot
lineEqualAreaPlot(lapply(Follins_angleAxis, function(s) c(s[1], s[2],
s[3])), shapes = '.')

# Plot a histogram of rotation amount between the northern domain and
the southern domain, in degrees.
hist(
     as.numeric(lapply(Follins_angleAxis, function(s)
        s[4] * 180 / pi)),
     30,
     xlab = "Angular Distance (degrees)",
     ylab = "Frequency",
     main = "Angular distances, Northern to Southern"
)

# Compute the mean and 1-sigma standard deviation of the rotation amount,
in Follins_angleAxisrees
mean(sapply(Follins_angleAxis, function(s) s[4] * 180 / pi))
sd(sapply(Follins_angleAxis, function(s) s[4] * 180 / pi))

# Compute the mean trend and plunge in Follins_angleAxisrees
Follins_axes <- lapply(Follins_angleAxis, function(s) c(s[1], s[2],
s[3]))
Follins_meanAxisDeg <-
geoTrendPlungeDegFromCartesian(lower(lineProjectedMean(Follins_axes)))

#Print the mean axis of rotation, in trend and plunge
Follins_meanAxisDeg

#==========================COMPARE FOLIATIONS FROM THE TWO
DATASETS========================================

# (Foliation-only vs. Foliation-lineation datasets)
```

```
# 1) Three plots to compare the foliations of the Fols (Foliation only
data set) and Follins (Foliation-lineation dataset) in the northern
domain

# Equal area plot of the Fols (cyan) and Follins (red)
lineEqualAreaPlotTwo(Follins$pole[Follins_northCrit],
Fols$pole[Fols_northCrit])

# Equal area plot of the bootstrapped means for the Fols (cyan) and
Follins (red)
lineEqualAreaPlotTwo(
                     FollinFols_northBoots$bootstraps,
                     Fols_northBoots$bootstraps
)

# Equal area plot of 95% confidence ellipses from bootstrapping for the
Fols (cyan) and Follins (red)
lineEqualAreaPlotTwo(
                     list(FollinFols_northBoots$center),
                     list(Fols_northBoots$center),
                     curves = list(FollinFols_northBoots$points,
                                   Fols_northBoots$points
                     )
)

#————————————————————————————————————————————————————————————

# 2) Three plots to compare the foliations of the Fols (Foliation only
data set) and Follins (Foliation-lineation dataset) in the central domain

# Equal area plot of the Fols (cyan) and Follins (red)
lineEqualAreaPlotTwo(
                     Follins$pole[Follins_centerCrit],
                     Fols$pole[Fols_centerCrit]
)

# Equal area plot of the bootstrapped means for the Fols (cyan) and
Follins (red)
lineEqualAreaPlotTwo(
                     FollinFols_centerBoots$bootstraps,
                     Fols_centerBoots$bootstraps
)

# Equal area plot of 95% confidence ellipses from bootstrapping for the
Fols (cyan) and Follins (red)
lineEqualAreaPlotTwo(
                     list(FollinFols_centerBoots$center),
                     list(Fols_centerBoots$center),
                     curves = list(FollinFols_centerBoots$points,
                                   Fols_centerBoots$points
                     )
)
```

```
#

# 3) Three plots to compare the foliations of the Fols (Foliation only
data set) and Follins (Foliation-lineation dataset) in the southern
domain

# Equal area plot of the Fols (cyan) and Follins (red)
lineEqualAreaPlotTwo(
                     Follins$pole[Follins_southCrit],
                     Fols$pole[Fols_southCrit]
)

# Equal area plot of the bootstrapped means for the Fols (cyan) and
Follins (red)
lineEqualAreaPlotTwo(
                     FollinFols_southBoots$bootstraps,
                     Fols_southBoots$bootstraps
)

# Equal area plot of 95% confidence ellipses from bootstrapping for the
Fols (cyan) and Follins (red)
lineEqualAreaPlotTwo(
                     list(FollinFols_southBoots$center),
                     list(Fols_southBoots$center),
                     curves = list(FollinFols_southBoots$points,
                                   Fols_southBoots$points
                     )
)


#===================================================================
#===================================================================



#==========================PART II: ORIENTATION STATISTICS ON FOLIATION-
LINEATION PAIRS========================================

#These are all foliation-lineation pairs, so they are orientation data.
We will proced with methods outlined in Davis and Titus, 2017.

#Since lines on foliation are bidirectional, these are line-in-plane
data, and have a four fold symmetry (see Davis and Titus, 2017).

#==========================PLOT                                    THE
DATA=======================================

# Plot the data in Equal area plot. Lineation (red), Pole to foliation
(cyan)
lineEqualAreaPlotTwo(
                     Follins$direction,
                     Follins$pole
)
```

```
# Plot the data in an Equal Volume Plot (after Davis and Titus, 2017).
Each point represents a foliation-lineation pair. (There are four copies
of the data due to mathematical symmetry)
oriEqualVolumePlot(
                  Follins$rotation,
                  oriLineInPlaneGroup
)

# We can also do some basic directional kamb contouring for poles and
lines. The numbers are the kamb intervals
lineKambPlot(
            Follins$pole,
            c(2, 4, 6, 8, 10, 12, 14, 18)
)
lineKambPlot(
            Follins$direction,
            c(2, 4, 6, 8, 10, 12, 14, 18)
)

# NOTE: This line requires c library to run--skip if you have not compiled
c.
# Uncomment lines below to Plot 6-sigma kamb contour in an Equal volume
plot of Follins.

# oricKambPlot(Follins$rotation,
#             group=oriLineInPlaneGroup,
#             multiple = 6,
#             simplePoints = TRUE,
#             backgroundColor="white", curveColor="black",
#             boundaryAlpha=0,
#             colors="black", axesColors=c("black", "black", "black"),
#             fogStyle="exp")

# If you wish to save this figure, maximize the plot window on your
screen before running this line. It will save to your working directory
folder.

#                    afterMaximizingWindow("WestMt_kamb_EqualVol1.png",
"WestMt_kamb_EqualVol2.png")


#==========================IDENTIFY                          GEOGRAPHIC
TRENDS========================================

# 1) Visually inspect the possibility of geographic trends in the data

# Plot an Equal Volume plot (after Davis and Titus, 2017), colored by
the three domains. Northern (red); Central (green); Southern (blue).
oriEqualVolumePlot(
                  Follins$rotation,
                  group = oriLineInPlaneGroup,
                  backgroundColor = "white",
```

```
                  curveColor = "black",
                  boundaryAlpha = 0.2,
                  colors = hues(Follins$domain),
                  axesColors = c("black", "black", "black")
)

# Plot an Equal volume plot of the data, colored by northing
oriEqualVolumePlot(
                  Follins$rotation,
                  group = oriLineInPlaneGroup,
                  backgroundColor = "white",
                  curveColor = "black",
                  boundaryAlpha = 0.2,
                  colors = hues(Follins$northing),
                  axesColors = c("black", "black", "black")
)

# Plot an Equal volume plot of the data, colored by easting
oriEqualVolumePlot(
                  Follins$rotation,
                  group = oriLineInPlaneGroup,
                  backgroundColor = "white",
                  curveColor = "black",
                  boundaryAlpha = 0.2,
                  colors = hues(Follins$easting),
                  axesColors = c("black", "black", "black")
)

# Save EqualVolumePlot figure. If you wish to save this figure, maximize
the plot window on your screen before running this line. It will save
to your working directory folder.
afterMaximizingWindow(
                     "WestMt_domains_EqualVol.png",
                     "WestMt_domains_EqualVol.png"
)

# Plot Follins in an Equal area plot, colored by domain. Lineation
(squares); Foliation (circles)
lineEqualAreaPlot(
                 c(Follins$pole, Follins$direction),
                 col = hues(Follins$domain),
                 shapes = c(replicate(length(Follins$pole), "c"),
                            replicate(length(Follins$direction), "s"))
)

#————————————————————————————————————————————————————————————

# 2) Run regressions to quantify any extant trends.

# 2A) Geodesic regression on all data. Perform a geodesic regression
with respect to azimuths every 10 degrees. Each regression asks the
question: "does orientation change linearly with respect to an azimuth
towards x degrees"
```

```
regressionSWestAll <- regressionSweep(Follins, 10, Follins_northCrit |
Follins_centerCrit | Follins_southCrit , 0)
westAllRegSum <- data.frame(regressionSWestAll[[19]])
names(westAllRegSum) <- c("Azimuth", "Error", "Min_Eigen", "R_squared",
"Pvalue")


# Plot the R^2 value (y-axis) as a function of the azimuth in degrees
(x-axis)
plot(
     as.vector(westAllRegSum[, 1]),
     as.vector(westAllRegSum[, 4]),
     main = "Geodesic regressions (all Domains)",
     xlab = "Azimuth in degrees",
     ylab = "R-squared"
)

#————————————————————————————————————————————————————————————

#2B) Perform geodesic regressions on each domain

# Perform a geodesic regression for the northern domain with respect to
azimuths every 10 degrees.
regressionSNorth <- regressionSweep(Follins, 10, Follins_northCrit, 0)
NorthRegSum <- data.frame(regressionSNorth[[19]])
names(NorthRegSum) = c("Azimuth", "Error", "Min_Eigen", "R_squared",
"Pvalue")

# Check that "Error" = 0, "Min_Eigen" always > 0.
NorthRegSum

# Plot the R^2 value (y-axis) as a function of the azimuth in degrees
(x-axis)
plot(
     as.vector(NorthRegSum$Azimuth),
     as.vector(NorthRegSum$R_squared),
     main = "Geodesic regressions (Northern Domain)",
     xlab = "Azimuth in degrees",
     ylab = "R-squared"
)

#————————————————————————————————————————————————————————————

# Perform a geodesic regression for the southern domain with respect to
azimuths every 10 degrees.
regressionSCenter <- regressionSweep(Follins, 10, Follins_centerCrit, 0)
CenterRegSum <- data.frame(regressionSCenter[[19]])
names(CenterRegSum) = c("Azimuth", "Error", "Min_Eigen", "R_squared",
"Pvalue")

# Check that "Error" = 0, "Min_Eigen" always > 0.
CenterRegSum
```

```
# Plot the R^2 value (y-axis) as a function of the azimuth in degrees
(x-axis)
plot(
     as.vector(CenterRegSum$Azimuth),
     as.vector(CenterRegSum$R_squared),
     main = "Geodesic regressions (Central Domain)",
     xlab = "Azimuth in degrees",
     ylab = "R-squared"
)

#————————————————————————————————————————————

# Perform a geodesic regression for the southern domain with respect to
azimuths every 10 degrees.
regressionSSouth <- regressionSweep(Follins, 10, Follins_southCrit, 0)
SouthRegSum <- data.frame(regressionSSouth[[19]])
names(SouthRegSum) = c("Azimuth", "Error", "Min_Eigen", "R_squared",
"Pvalue")

# Check that "Error" = 0, "Min_Eigen" always > 0.
SouthRegSum

# Plot the R^2 value (y-axis) as a function of the azimuth in degrees
(x-axis)
plot(
     as.vector(SouthRegSum$Azimuth),
     as.vector(SouthRegSum$R_squared),
     main = "Geodesic regressions (Southern Domain)",
     xlab = "Azimuth in degrees",
     ylab = "R-squared"
)

#————————————————————————————————————————————

# 3) Perform a Kernal regression for all Follins.

# Recategorize the symmetric copies of the data.
mu <- oriFrechetMean(Follins$rotation, group=oriLineInPlaneGroup)
Follins$rotation <- oriNearestRepresentatives(Follins$rotation, mu,
group=oriLineInPlaneGroup)

# A precomputed value for bandwidth--use this value to save time.

bandwidth <- 0.5879134

# Uncomment line below to compute the appropriate bandwidth for Kernal
regression with respect to easting.
bandwidth                                                              <-
rotBandwidthForKernelRegression(Follins$easting,Follins$rotation,
dnorm)

#Run the kernel regression to generate a bunch of points.There were no
errors and all minEigenvalues were >0
```

```
kernelReg <- lapply(
                    seq(from = min(Follins$easting), to =
max(Follins$easting), by = 100),
                    rotKernelRegression,
                    Follins$easting,
                    Follins$rotation,
                    bandwidth,
                    numSteps = 1000
)
sapply(kernelReg, function(regr) regr$error)
sapply(kernelReg, function(regr) {regr$minEigenvalue > 0})

# Plot the regression curve.
kernelRegCurve <- lapply(kernelReg, function(regr) regr$r)
oriEqualVolumePlot(
                   kernelRegCurve,
                   simplePoints = TRUE,
                   oriLineInPlaneGroup
)

# Compute the R^2.
KernRsquared <- rotRsquaredForKernelRegression(Follins$easting,
Follins$rotation, bandwidth, numSteps =
                                       1000)
KernRsquared

# Do a permutation test for significance. For the sake of time, we do
only 10 permutations, although that is far too few to tell us anything.
rSquareds <- rotKernelRegressionPermutations(Follins$easting,
Follins$rotation, bandwidth, numPerms =
                                        1000)
length(rSquareds)
p <- sum(rSquareds > KernRsquared$rSquared)
p


#=========================STATISTICAL
DESCRIPTORS========================================

# 1) Northern Domain, n = 16

# Frechet mean (minimizes the Frechet variance).
FrechetMeanNorth <- oriFrechetMean(Follins$rotation[Follins_northCrit],
oriLineInPlaneGroup)
FrechetVarNorth <- oriVariance(Follins$rotation[Follins_northCrit],
FrechetMeanNorth, oriLineInPlaneGroup)

# Plot the FrechetMean in an Equal Volume plot
FrechetCurvesNorth <- lapply(Follins$rotation[Follins_northCrit],
function(r) rotGeodesicPoints(FrechetMeanNorth, r, 10))
rotEqualAnglePlot(points = Follins$rotation[Follins_northCrit], curves
= FrechetCurvesNorth)
```

```
# Print the Strike, Dip, Rake of the Frechet mean.
geoStrikeDipRakeDegFromRotation(FrechetMeanNorth)

# Print the Frechet variance.
FrechetVarNorth

#------------------------------------------------------------

# 2) Central Domain, n = 34
#Frechet mean minimizes the Frechet variance.
FrechetMeanCenter                                         <-
oriFrechetMean(Follins$rotation[Follins_centerCrit],
oriLineInPlaneGroup)
FrechetVarCenter  <-  oriVariance(Follins$rotation[Follins_centerCrit],
FrechetMeanCenter, oriLineInPlaneGroup)

#plot the FrechetMean
FrechetCurvesCenter   <-   lapply(Follins$rotation[Follins_centerCrit],
function(r) rotGeodesicPoints(FrechetMeanCenter, r, 10))
rotEqualAnglePlot(
                  points = Follins$rotation[Follins_centerCrit],
                  curves = FrechetCurvesCenter
)

# Print the Strike, Dip, Rake of the Frechet mean.
geoStrikeDipRakeDegFromRotation(FrechetMeanCenter)

# Print the Frechet variance.
FrechetVarCenter

#------------------------------------------------------------

# 3) Southern Domain, n = 79
#Frechet mean minimizes the Frechet variance.
FrechetMeanSouth <- oriFrechetMean(Follins$rotation[Follins_southCrit],
oriLineInPlaneGroup)
FrechetVarSouth  <-    oriVariance(Follins$rotation[Follins_southCrit],
FrechetMeanSouth, oriLineInPlaneGroup)

#plot the FrechetMean
FrechetCurvesSouth    <-    lapply(Follins$rotation[Follins_southCrit],
function(r) rotGeodesicPoints(FrechetMeanSouth, r, 10))
rotEqualAnglePlot(points = Follins$rotation[Follins_southCrit], curves
= FrechetCurvesSouth)

# Print the Strike, Dip, Rake of the Frechet mean.
geoStrikeDipRakeDegFromRotation(FrechetMeanSouth)

# Print the Frechet variance.
FrechetVarSouth

#------------------------------------------------------------
```

```
# 4) Dispersion of the data

#The standard way to get at dispersion is to compute the maximum matrix
fisher likelihood. The matrix fisher distribution comprises a kind of
"mean" and a positive definite symmetric matrix, which characterises the
anisotropic dispersion.

# Northern Domain, n = 16. Fisher maximum likelihood
mleNorth <- rotFisherMLE(Follins$rotation[Follins_northCrit])
mleNorth
eigen(mleNorth$kHat, symmetric = TRUE, only.value = TRUE)$values

#Central Domain, n = 34. Fisher maximum likelihood
mleCenter <- rotFisherMLE(Follins$rotation[Follins_centerCrit])
mleCenter
eigen(mleCenter$kHat, symmetric = TRUE, only.value = TRUE)$values

#Southern Domain, n = 79. Fisher maximum likelihood
mleSouth <- rotFisherMLE(Follins$rotation[Follins_southCrit])
mleSouth
eigen(mleSouth$kHat, symmetric = TRUE, only.value = TRUE)$values



#==========================INFERENCE + HYPOTHESIS
TESTING========================================

# Here, we can perform some statistical techniques to use information
about the samples (each domain) to compute a probability cloud of the
mean of the population(s) from which those samples were taken.
# From numerical work in Davis and Titus (2017), MCMC will work the best
for small sample sizes that have the matrix fisher anisotropy
(eigenvalues) of (large, large, small). All four domains have this
anisotropy, and have too few data points to use the (computationally
quicker) bootstrapping method.

#We'll do both and compare.

# Compute the bootstrap cloud of means
Follins_northBoots <-
oriBootstrapInference(Follins$rotation[Follins_northCrit], 10000,
oriLineInPlaneGroup)
Follins_centerBoots <-
oriBootstrapInference(Follins$rotation[Follins_centerCrit], 10000,
oriLineInPlaneGroup)
Follins_southBoots <-
oriBootstrapInference(Follins$rotation[Follins_southCrit], 10000,
oriLineInPlaneGroup)

# Compute the Markov chain Monte Carlo cloud of means
northMCMC <-
oricWrappedTrivariateNormalMCMCInference(Follins$rotation[Follins_nort
hCrit],
```

```
                                                      group          =
oriLineInPlaneGroup,
                                                      numCollection   =
100
              )
centerMCMC                                                           <-
oricWrappedTrivariateNormalMCMCInference(Follins$rotation[Follins_cent
erCrit],
                                                       group          =
oriLineInPlaneGroup,
                                                       numCollection  =
100
              )
southMCMC                                                            <-
oricWrappedTrivariateNormalMCMCInference(Follins$rotation[Follins_sout
hCrit],
                                                       group          =
oriLineInPlaneGroup,
                                                       numCollection  =
100
              )

#————————————————————————————————————————————————————————————

# 1) Northern v. Central domains

# A) Using MCMC

# Construct the 95% confidence ellipsoids from small triangles
trisNorthMCMC <-  rotEllipsoidTriangles(northMCMC$mBar,
                                        northMCMC$leftCovarInv,
                                        northMCMC$q095,
                                        numNonAdapt = 4)
trisCenterMCMC <- rotEllipsoidTriangles(centerMCMC$mBar,
                                        centerMCMC$leftCovarInv,
                                        centerMCMC$q095,
                                        numNonAdapt = 4)

# Plot the MCMC comparison in an equal Volume plot, with 95% confidence
ellipsoids
oriEqualAnglePlot(
                  points = c(northMCMC$ms, centerMCMC$ms),
                  boundaryAlpha = .1,
                  axesColors = c("black", "black", "black"),
                  fogStyle = "none",
                  background = "white",
                  triangles = c(trisNorthMCMC, trisCenterMCMC),
                  simplePoints = TRUE,
                  colors = c(replicate(length(northMCMC$ms), "black"),
replicate(length(centerMCMC$ms), "orange")),
                  group = oriTrivialGroup
)
```

```
# If you wish to save this figure, maximize the plot window on your screen before running this line. It will save to your working directory folder.
afterMaximizingWindow("MCMC_WestMt_NC_1.png", "MCMC_WestMt_NC_2.png")

# Plot the MCMC comparison in an Equal Area plot
lineEqualAreaPlotTwo(c(
    lapply(northMCMC$ms, function(s)
        s[1,]),
    lapply(northMCMC$ms, function(s)
        s[2,])
),
c(
    lapply(centerMCMC$ms, function(s)
        s[1,]),
    lapply(centerMCMC$ms, function(s)
        s[2,])
))

#....................................................................

# B) Using bootstrapping

# Construct the 95% confidence ellipsoids from small triangles
trisNorthBoot <-
    rotEllipsoidTriangles(
        Follins_northBoots$center,
        Follins_northBoots$leftCovarInv,
        Follins_northBoots$q095,
        numNonAdapt = 4
    )
trisCenterBoot <-
    rotEllipsoidTriangles(
        Follins_centerBoots$center,
        Follins_centerBoots$leftCovarInv,
        Follins_centerBoots$q095,
        numNonAdapt = 4
    )

# Plot the bootstrap comparison in an equal Volume plot, with 95% confidence ellipsoids
rotEqualAnglePlot(
    points = c(
        Follins_northBoots$bootstraps,
        Follins_centerBoots$bootstraps
    ),
    triangles = c(trisNorthBoot, trisCenterBoot),
    boundaryAlpha = .1,
    axesColors = c("black", "black", "black"),
    fogStyle = "none",
    background = "white",
    simplePoints = TRUE,
    colors = c(replicate(
```

```
        length(Follins_northBoots$bootstraps), "black"
    ), replicate(
        length(Follins_centerBoots$bootstraps), "orange"
    ))
)

# If you wish to save this figure, maximize the plot window on your screen before running this line. It will save to your working directory folder.
afterMaximizingWindow("Boots_WestMt_NC_1.png", "Boots_WestMt_NC_2.png")

# Plot the bootstrap comparison in an Equal Area plot
lineEqualAreaPlotTwo(c(
    lapply(Follins_northBoots$bootstraps, function(s)
        s[1,]),
    lapply(Follins_northBoots$bootstraps, function(s)
        s[2,])
),
c(
    lapply(Follins_centerBoots$bootstraps, function(s)
        s[1,]),
    lapply(Follins_centerBoots$bootstraps, function(s)
        s[2,])
))

#————————————————————————————————————————————————————————————

# 2) Northern vs. Southern domains

# A) Using MCMC

# Construct the 95% confidence ellipsoids from small triangles
trisNorthMCMC <-
    rotEllipsoidTriangles(northMCMC$mBar,
                          northMCMC$leftCovarInv,
                          northMCMC$q095,
                          numNonAdapt = 5)
trisSouthMCMC <-
    rotEllipsoidTriangles(southMCMC$mBar,
                          southMCMC$leftCovarInv,
                          southMCMC$q095,
                          numNonAdapt = 5)

# Plot the bootstrap comparison in an equal Volume plot, with 95% confidence ellipsoids
oriEqualAnglePlot(
    points = c(northMCMC$ms, southMCMC$ms),
    triangles = c(trisNorthMCMC, trisSouthMCMC),
    boundaryAlpha = 0.1,
    axesColors = c("black", "black", "black"),
    fogStyle = "none",
    background = "white",
    simplePoints = TRUE,
```

```
    colors      =      c(replicate(length(northMCMC$ms),      "black"),
replicate(length(southMCMC$ms), "blue")),
    group = oriTrivialGroup
)

# If you wish to save this figure, maximize the plot window on your
screen before running this line. It will save to your working directory
folder.
afterMaximizingWindow("MCMC_WestMt_NS_1.png", "MCMC_WestMt_NS_2.png")

# Plot the MCMC comparison in an Equal Area plot
lineEqualAreaPlotTwo(c(
    lapply(northMCMC$ms, function(s)
        s[1,]),
    lapply(northMCMC$ms, function(s)
        s[2,])
),
c(
    lapply(southMCMC$ms, function(s)
        s[1,]),
    lapply(southMCMC$ms, function(s)
        s[2,])
))

#.....................................................................

# B) Using bootstrapping
# Construct the 95% confidence ellipsoids from small triangles
trisNorthBoot <-
    rotEllipsoidTriangles(
        Follins_northBoots$center,
        Follins_northBoots$leftCovarInv,
        Follins_northBoots$q095,
        numNonAdapt = 4
    )
trisSouthBoot <-
    rotEllipsoidTriangles(
        Follins_southBoots$center,
        Follins_southBoots$leftCovarInv,
        Follins_southBoots$q095,
        numNonAdapt = 4
    )

# Plot the bootstrap comparison in an equal Volume plot, with 95%
confidence ellipsoids.
rotEqualAnglePlot(
    points = c(
        Follins_northBoots$bootstraps,
        Follins_southBoots$bootstraps
    ),
    triangles = c(trisNorthBoot, trisSouthBoot),
    boundaryAlpha = .1,
    axesColors = c("black", "black", "black"),
```

```
    fogStyle = "none",
    background = "white",
    simplePoints = TRUE,
    colors = c(replicate(
        length(Follins_northBoots$bootstraps), "black"
    ),
    replicate(
        length(Follins_southBoots$bootstraps), "blue"
    ))
)

# If you wish to save this figure, maximize the plot window on your
screen before running this line. It will save to your working directory
folder.
afterMaximizingWindow("Boots_WestMt_NS_1.png", "Boots_WestMt_NS_2.png")

# Plot the bootstrap comparison in an Equal Area plot
lineEqualAreaPlotTwo(c(
    lapply(Follins_northBoots$bootstraps, function(s)
        s[1,]),
    lapply(Follins_northBoots$bootstraps, function(s)
        s[2,])
),
c(
    lapply(Follins_southBoots$bootstraps, function(s)
        s[1,]),
    lapply(Follins_southBoots$bootstraps, function(s)
        s[2,])
))



#————————————————————————————————————————————————————————————

# 3) Central vs. Southern domains

# A) Using MCMC

# Construct the 95% confidence ellipsoids from small triangles
trisCenterMCMC <-
    rotEllipsoidTriangles(centerMCMC$mBar,
                          centerMCMC$leftCovarInv,
                          centerMCMC$q095,
                          numNonAdapt = 4)
trisSouthMCMC <-
    rotEllipsoidTriangles(southMCMC$mBar,
                          southMCMC$leftCovarInv,
                          southMCMC$q095,
                          numNonAdapt = 4)

# Plot the bootstrap comparison in an equal Volume plot, with 95%
confidence ellipsoids
oriEqualAnglePlot(
```

```
    points = c(centerMCMC$ms, southMCMC$ms),
    triangles = c(trisCenterMCMC, trisSouthMCMC),
    boundaryAlpha = 0.1,
    axesColors = c("black", "black", "black"),
    fogStyle = "none",
    background = "white",
    simplePoints = TRUE,
    colors = c(replicate(length(centerMCMC$ms), "orange"), replicate(length(southMCMC$ms), "blue")),
    group = oriTrivialGroup
)

# If you wish to save this figure, maximize the plot window on your screen before running this line. It will save to your working directory folder.
afterMaximizingWindow("MCMC_WestMt_CS_1.png", "MCMC_WestMt_CS_2.png")

# Plot the MCMC comparison in an Equal Area plot
lineEqualAreaPlotTwo(c(
    lapply(centerMCMC$ms, function(s)
        s[1,]),
    lapply(centerMCMC$ms, function(s)
        s[2,])
),
c(
    lapply(southMCMC$ms, function(s)
        s[1,]),
    lapply(southMCMC$ms, function(s)
        s[2,])
))

#.....................................................................


# B) Using bootstrapping

# Construct the 95% confidence ellipsoids from small triangles
trisCenterBoot <-
    rotEllipsoidTriangles(
        Follins_centerBoots$center,
        Follins_centerBoots$leftCovarInv,
        Follins_centerBoots$q095,
        numNonAdapt = 4
    )
trisSouthBoot <-
    rotEllipsoidTriangles(
        Follins_southBoots$center,
        Follins_southBoots$leftCovarInv,
        Follins_southBoots$q095,
        numNonAdapt = 4
    )
```

```
# Plot the bootstrap comparison in an equal Volume plot, with 95% confidence ellipsoids
oriEqualAnglePlot(
    points = c(
        Follins_centerBoots$bootstraps,
        Follins_southBoots$bootstraps
    ),
    triangles = c(trisCenterBoot, trisSouthBoot),
    boundaryAlpha = 0.1,
    axesColors = c("black", "black", "black"),
    fogStyle = "none",
    background = "white",
    simplePoints = TRUE,
    colors = c(replicate(
        length(Follins_northBoots$bootstraps), "orange"
    ), replicate(
        length(Follins_southBoots$bootstraps), "blue"
    )),
    group = oriTrivialGroup
)

# If you wish to save this figure, maximize the plot window on your screen before running this line. It will save to your working directory folder.
afterMaximizingWindow("Boots_WestMt_CS_1.png", "Boots_WestMt_CS_2.png")

# Plot the bootstrap comparison in an Equal Area plot
lineEqualAreaPlotTwo(c(
    lapply(Follins_centerBoots$bootstraps, function(s)
        s[1,]),
    lapply(Follins_centerBoots$bootstraps, function(s)
        s[2,])
),
c(
    lapply(Follins_southBoots$bootstraps, function(s)
        s[1,]),
    lapply(Follins_southBoots$bootstraps, function(s)
        s[2,])
))
```

APPENDIX 5: Ahsahka segment data

Ahsahka segment foliation-lineation pairs

location	easting	northing	strike	dip	trend	plunge
13-TSL-32	553090	5153114	288	53	30	58
13-TSL-20	553117	5153753	288	62	55	58
12-TSL-15	553121	5152301	300	56	24	55
13-TSL-31	553359	5152857	308	56	355	52
13-TSL-7	553379	5154706	295	63	313	29
12-TSL-8	553387	5152784	297	48	16	47

13-TSL-39	553464	5154349	294	66	346	65
13-TSL-6	553548	5154710	315	60	324	27
12-TSL-24	554220	5159474	288	50	41	48
12-TSL-29	554942	5160955	284	52	62	41
12-TSL-19	554947	5155847	286	59	337	52
13-TSL-4	554979	5155062	292	56	13	63
12-TSL-31	555105	5160561	285	62	50	57
12-TSL-9	555453	5149141	295	54	53	50
13-TSL-37.1	555527	5150617	323	63	53	63
13-TSL-36	555543	5152055	309	75	350	53
12-TSL-33	555631	5160286	264	71	61	48
12-TSL-36	556012	5157321	303	52	9	49
13-TSL-10.1	556042	5157269	247	48	18	40
13-TSL-1	556201	5156192	315	51	24	48
13-TSL-38.1	556490	5154729	302	54	332	48
12-TSL-21	556647	5156041	303	52	355	45
13-TSL-14	556812	5156241	293	69	348	58
13-TSL-50.4	557395	5164863	52	67	71	39
13-TSL-27	557447	5155990	309	66	22	68
13-TSL-25	557532	5156097	306	51	19	52
13-TSL-41	557800	5163786	74	31	238	16
13-TSL-24	557836	5155990	327	61	5	42
13-TSL-23.1	557969	5155993	330	78	349	57
13-TSL-42	557971	5163945	39	38	60	9
13-TSL-43	558190	5164262	357	30	28	16
13-TSL-49.9	558304	5164892	206	42	214	5
13-TSL-46	558615	5169780	326	49	67	47
13-TSL-44.3	558729	5164569	21	41	41	15
13-TSL-44.32	558745	5164569	44	58	39	4
13-TSL-44.1	558786	5164589	15	33	53	12
13-TSL-49.4	558956	5168786	336	59	61	55
13-TSL-45.4	558972	5165164	290	61	70	49
13-TSL-49.5	558982	5168403	319	45	49	45
13-TSL-49	559063	5170431	317	38	33	35
13-TSL-49.1	559075	5170416	22	35	69	25
13-TSL-49.2	559077	5170414	335	52	65	52
13-TSL-49.3	559079	5170412	323	61	51	59
13-TSL-49.6	559117	5167347	311	55	70	51
13-TSL-46.1	559163	5170086	320	57	55	54
13-TSL-45.9	559193	5167930	327	50	46	45

13-TSL-45.8	559400	5167731	309	50	78	44
13-TSL-49.7	559479	5165587	325	34	37	33
13-TSL-48.2	559485	5170232	300	76	59	66
12-TSL-14	559665	5163392	279	30	48	23
13-TSL-44.6	559887	5165513	350	50	66	46
13-TSL-47i	559899	5170340	292	83	94	73
13-TSL-48	559987	5170419	283	86	56	68
13-TSL-45.7	560015	5167128	344	49	36	41
13-TSL-45.2	560039	5166529	11	30	48	19
13-TSL-52.2	561482	5161721	312	46	35	43
13-TSL-52.4	562182	5161385	329	40	30	36
13-TSL-52.3	562183	5161384	341	39	71	38
13-TSL-53.2	562445	5161179	335	35	69	34
13-TSL-53	562469	5161107	308	44	47	42
13-TSL-53.4	562495	5161175	318	49	59	47
13-TSL-54.1	563347	5160922	325	55	70	57
13-TSL-54.2	563387	5160922	305	58	94	39
13-TSL-55.1	563640	5161621	349	35	41	30
12-TSL-11	563916	5160475	300	60	79	48
13-TSL-58.1	564407	5161525	316	89	315	6
13-TSL-58	566295	5161832	326	61	74	57
13-TSL-57.1	566475	5161692	327	58	29	58
13-TSL-56.8	567079	5161942	347	58	7	33

APPENDIX 6: **Ahsahka segment statistical analysis**

```
### PRELIMINARY WORK ####
# Set the working directory.
setwd("~/Desktop/20170620geologyGeometry")
# Load the necessary R libraries.
source("library/all.R")
# NOTE: Run the following line only if you have compiled C. Markov chain
Monte Carlo requires C. Load the necessary library
source("libraryC/all.R")
source("JSG_statsFunctions.r")

#==========================LOAD                                    THE
DATA========================================

# Load the foliation-lineation data
Follins <- geoDataFromFile("data/Follins_Ahs.csv")

# Check how many measurements there are
nrow(Follins)
```

```
#==========================PLOT                                     THE
DATA========================================

# Plot foliation-lineation locations in map view
plot(
    Follins$easting,
    Follins$northing,
    xlab = "Easting (meters)",
    ylab = "Northing (meters)"
)

# Plot the data in Equal area plot. Pole to foliation (red), Lineation
(cyan)
lineEqualAreaPlotTwo(lapply(Follins$rotation, function(s)
    s[1, ]),
    lapply(Follins$rotation, function(s)
        s[2, ]))

# Plot Follins in Equal Area plots with Kamb contours (numbers are 2
sigma values)
# Poles to foliation (circles)
lineKambPlot(lapply(Follins$rotation, function(s)
    s[1, ]), c(2, 6, 10, 14, 18))
# Lineations (squares)
lineKambPlot(lapply(Follins$rotation, function(s)
    s[2, ]),
    c(2, 6, 10, 14, 18),
    shapes = c("s"))

# Plot the Follins in an Equal Volume plot (after Davis and Titus, 2017)
oriEqualVolumePlot(
    Follins$rotation,
    group = oriLineInPlaneGroup,
    backgroundColor = "white",
    curveColor = "black",
    boundaryAlpha = 0.1,
    colors = "black" ,
    axesColors = c("black", "black", "black"),
    fogStyle = "none"
)

# If you wish to save this figure, maximize the plot window on your
screen before running this line. It will save to your working directory
folder.
afterMaximizingWindow("Ahsahka_EqualVol1.png", "Ahsahka_EqualVol2.png")

# NOTE: Requires C. Skip if you have not compiled C. Plot 6-sigma Kamb
contours for the data in an equal volume plot (after Davis and Titus,
2017)
oricKambPlot(
    Follins$rotation,
    group = oriLineInPlaneGroup,
    multiple = 6,
```

```
    backgroundColor = "white",
    curveColor = "black",
    boundaryAlpha = 0.1,
    colors = "black",
    axesColors = c("black", "black", "black"),
    fogStyle = "none"
)

# If you wish to save this figure, maximize the plot window on your screen before running this line. It will save to your working directory folder.
afterMaximizingWindow("Ahsahka_Kamb_equalVol1.png",
"Ahsahka_Kamb_EqualVol2.png")


#=========================IDENTIFY GEOGRAPHIC TRENDS=========================================

# 1) Visually inspect the possibility of geographic trends in the data

# Plot foliation-lineation locations in map view, colored in gray-scale by northing. This shading will be used in the following plots.
plot(
    Follins$easting,
    Follins$northing,
    col = shades(Follins$northing),
    pch = 19
)

# Plot foliation-lineation orientations in an equal volume plot (after Davis and Titus, 2017)
oriEqualVolumePlot(
    Follins$rotation,
    group = oriLineInPlaneGroup,
    simplePoints = FALSE,
    backgroundColor = "white",
    curveColor = "black",
    boundaryAlpha = 0.1,
    colors = shades(Follins$northing),
    axesColors = c("black", "black", "black"),
    fogStyle = "none"
)

# If you wish to save this figure, maximize the plot window on your screen before running this line. It will save to your working directory folder.
afterMaximizingWindow("Ahsahka_shadesNorthing_equalVol1.png",
                      "Ahsahka_shadesNorthing_EqualVol2.png")

# Plot foliation-lineation orientations colored by northing. Poles to foliation (circles), Lineation (squares)
lineEqualAreaPlot(
    c(
```

```
        lapply(Follins$rotation, function(s)
            s[1, ]),
        lapply(Follins$rotation, function(s)
            s[2, ])
    ),
    col = shades(Follins$northing),
    shapes = c(replicate(length(Follins$rotation), "c"),
replicate(length(Follins$rotation), "s"))
)

#————————————————————————————————————————————————————————————

# 2) Run geodesic regressions to quantify any extant trends.

# Create a temporary version of the foliation lineation data and
establish a geographic criteria that encompasses all the data.
FollinsAll <- Follins
AhsAllCrit <-
    FollinsAll$easting > 550000 & Follins$northing > 5140000
FollinsAll$domain <- replicate(nrow(FollinsAll), 1)
Follins$domain[AhsAllCrit] <- 1

# 2A) Perform a geodesic regression with respect to azimuths every 10
degrees on all data, using the geographic criteria defined above. Each
regression asks the question: "does orientation change linearly with
respect to an azimuth towards x degrees"
regressions <- regressionSweep(FollinsAll, 10, (AhsAllCrit), 0)

# Create a data frame that contains the summary information for the
regressions
regressionsSum <- data.frame(regressions[[19]])

# Name the columns of the data frame
names(regressionsSum) = c("Azimuth", "Error", "Min_Eigen", "R_squared",
"Pvalue")

# View the summary table for the output.
regressionsSum

# Plot the R^2 value as a function of Azimuth
plot(
    as.vector(regressionsSum$Azimuth),
    as.vector(regressionsSum$R_squared),
    xlab = "Azimuth",
    ylab = "R-squared"
)

#————————————————————————————————————————————————————————————

# 3) Run kernal regression to quantify any extant trends.

# Recategorize the symmetric copies of the data.
mu <- oriFrechetMean(Follins$rotation, group=oriLineInPlaneGroup)
```

```
Follins$rotation <- oriNearestRepresentatives(Follins$rotation, mu,
group=oriLineInPlaneGroup)

# Define the bandwith for a kernal regression. If desired, uncomment the
next line (it may take ~20 minutes to run). Otherwise, use the pre-
calculated value, "bandwidth".
bandwidth <-
rotBandwidthForKernelRegression(Follins$northing,Follins$rotation)
bandwidth <- 0.02314421

# Perform a kernel regression with respect to northing. The kernel
function is analogous to finding a best fit curve in 2D.
kernelReg <- lapply(
    seq(
        from = min(Follins$northing),
        to = max(Follins$northing),
        by = 100
    ),
    rotKernelRegression,
    Follins$northing,
    Follins$rotation,
    bandwidth,
    numSteps = 1000
)

# Filter the regression results to only keep those that give an error
of 0 and minimum eigen value > 0
sapply(kernelReg, function(regr)
    regr$error)
sapply(kernelReg, function(regr) {
    regr$minEigenvalue > 0
})


# Plot the regression curve in an equal volume plot.
kernelRegCurve <- lapply(kernelReg, function(regr)
    regr$r)
oriEqualVolumePlot(
    kernelRegCurve,
    group=oriLineInPlaneGroup,
    backgroundColor = "white",
    curveColor = "black",
    boundaryAlpha = 0.1,
    colors = "black",
    axesColors = c("black", "black", "black"),
    fogStyle = "none",
    simplePoints = TRUE
)

# Compute the R^2 for the kernel regression.
KernRsquared <-
    rotRsquaredForKernelRegression(Follins$northing, Follins$rotation,
bandwidth, numSteps =
```

```
                                        1000)

# View the R^2 value
KernRsquared

# Do a permutation test for significance. This takes a significant amount
of time (>>1 hr)
rSquareds <-
    rotKernelRegressionPermutations(Follins$northing, Follins$rotation,
bandwidth, numPerms =
                                        1000)
length(rSquareds)
p <- sum(rSquareds > KernRsquared$rSquared)
p


#==========================DEFINE GEOGRAPHIC
DOMAINS========================================

# Define geographic criteria, based on sampling area, with the proposed
shear zone boundary into account.
domain1Crit <- Follins$easting < 560000 & Follins$northing < 5158000
domain2Crit <- (Follins$easting >= 560000 & Follins$northing < 5163000)
| Follins$easting >= 565000
domain3Crit <- !(domain1Crit | domain2Crit)

# # Create a new column in the dataframe in which to store the domain
information
Follins$domain <- replicate(nrow(Follins), 1)

# Classify the foliation-only dataset by domain
Follins$domain[domain1Crit] <- 1
Follins$domain[domain2Crit] <- 2
Follins$domain[domain3Crit] <- 3

# Plot the locations of the foliation-lineation data in map view, each
domain a different color. Domain 1 (Red), Domain 2 (Green), Domain 3
(Blue)
plot(
    x = Follins$easting,
    y = Follins$northing,
    xlab = "Easting (meters)",
    ylab = "Northing (meters)",
    col = hues(Follins$domain),
    pch = 19
)

#==========================IDENTIFY GEOGRAPHIC TRENDS WITHIN
DOMAINS========================================

# 1) Make sure each domain is roughly unimodal
#DOMAIN 1, n = 23.
oriEqualVolumePlot(
```

```
    Follins$rotation[domain1Crit],
    oriLineInPlaneGroup,
    backgroundColor = "white",
    curveColor = "black",
    boundaryAlpha = 0.1,
    colors = "black",
    axesColors = c("black", "black", "black"),
    fogStyle = "none"
)
oricKambPlot(
    Follins$rotation[domain1Crit],
    oriLineInPlaneGroup,
    backgroundColor = "white",
    curveColor = "black",
    boundaryAlpha = 0.1,
    colors = "black",
    axesColors = c("black", "black", "black"),
    fogStyle = "none"
)

#DOMAIN 2, n = 14.
oriEqualVolumePlot(
    Follins$rotation[domain2Crit],
    oriLineInPlaneGroup,
    backgroundColor = "white",
    curveColor = "black",
    boundaryAlpha = 0.1,
    colors = "black",
    axesColors = c("black", "black", "black"),
    fogStyle = "none"
)
oricKambPlot(
    Follins$rotation[domain2Crit],
    oriLineInPlaneGroup,
    backgroundColor = "white",
    curveColor = "black",
    boundaryAlpha = 0.1,
    colors = "black",
    axesColors = c("black", "black", "black"),
    fogStyle = "none"
)

#DOMAIN 3, n = 32.
oriEqualVolumePlot(
    Follins$rotation[domain2Crit],
    oriLineInPlaneGroup,
    backgroundColor = "white",
    curveColor = "black",
    boundaryAlpha = 0.1,
    colors = "black",
    axesColors = c("black", "black", "black"),
    fogStyle = "none"
)
```

```
oriEqualVolumePlot(Follins$rotation[domain2Crit],
                   oriLineInPlaneGroup,
                   col = hues(Follins$easting[domain2Crit]))
oricKambPlot(
    Follins$rotation[domain2Crit],
    oriLineInPlaneGroup,
    backgroundColor = "white",
    curveColor = "black",
    boundaryAlpha = 0.1,
    colors = "black",
    axesColors = c("black", "black", "black"),
    fogStyle = "none"
)


# Plot all the foliation data in an equal volume plot (after Davis and
Titus, 2017), colored by domain. Domain 1 (red), Domain 2 (green), Domain
3 (blue)
oriEqualVolumePlot(
    Follins$rotation,
    oriLineInPlaneGroup,
    col = hues(Follins$domain),
    backgroundColor = "white",
    curveColor = "black",
    boundaryAlpha = 0.2,
    axesColors = c("black", "black", "black"),
    fogStyle = "none"
)

#------------------------------------------------------------

# 2) Look for geographic patterns in plots.
#DOMAIN 1, n = 23. maybe a very faint rainbow--more like clumping of the
dark blues together.
oriEqualVolumePlot(Follins$rotation[domain1Crit],
                   oriLineInPlaneGroup,
                   col = shades(Follins$northing[domain1Crit]))

#DOMAIN 2, n = 14. Part of the problem with small sample sizes--you can't
really know whether the datapoints are independent...
oriEqualVolumePlot(Follins$rotation[domain2Crit],
                   oriLineInPlaneGroup,
                   col = shades(Follins$northing[domain2Crit]))

#DOMAIN 3, n = 32. No rainbow.
oriEqualVolumePlot(Follins$rotation[domain2Crit],
                   oriLineInPlaneGroup,
                   col = shades(Follins$easting[domain2Crit]))

#------------------------------------------------------------
```

```
# 3) From visual inspection of the above plots, Domain 1 and Domain 3
potentially have geographic dependency. Perform regressions on these two
domains.

#DOMAIN 1, n = 23. Perform a series of Geodesic Regressions for every
10° azimuth, with p-values not calculated.
regressionsD1 <- regressionSweep(Follins, 10, domain1Crit, 0)

# Create a dataframe with the summary information for the regressions
regressionsD1Sum <- data.frame(regressionsD1[[19]])

# Name the columns of the data frame
names(regressionsD1Sum) = c("Azimuth", "Error", "Min_Eigen",
"R_squared", "Pvalue")

# View the summary table for the output.
regressionsD1Sum

# Plot the R^2 value as a function of Azimuth
plot(
    as.vector(regressionsD1Sum$Azimuth),
    as.vector(regressionsD1Sum$R_squared),
    xlab = "Azimuth",
    ylab = "R-squared"
)

#————————————————————————————————————————————————————————————

# Define the Azimuth with the highest R-squared value. (In this case,
130)
azimuthD1 <-
    Follins$easting[domain1Crit] * sin(130 * degree) +
Follins$northing[domain1Crit] *
    cos(130 * degree)

# DOMAIN 1, n = 23. Define the bandwith for a kernal regression. If
desired, uncomment the next line (it may take ~20 minutes to run).
Otherwise, use the pre-calculated value, "bandwidthD1".
# bandwidthD1 <-
rotBandwidthForKernelRegression(azimuthD1,Follins$rotation[domain1Crit
])
bandwidthD1 <- 0.02202349

#Run the kernel regression
kernelRegD1 <- lapply(
    seq(
        from = min(azimuthD1),
        to = max(azimuthD1),
        by = 100
    ),
    rotKernelRegression,
    azimuthD1,
    Follins$rotation[domain1Crit],
```

```
    bandwidthD1,
    numSteps = 1000
)
sapply(kernelRegD1, function(regr)
    regr$error)
sapply(kernelRegD1, function(regr) {
    regr$minEigenvalue > 0
})

# Plot the regression curve
kernelRegCurveD1 <- lapply(kernelRegD1, function(regr)
    regr$r)
oriEqualVolumePlot(Follins$rotation[domain1Crit],
                   curves = list(kernelRegCurveD1),
                   simplePoints = TRUE, oriLineInPlaneGroup)

# Compute the R^2.
KernRsquaredD1 <-
    rotRsquaredForKernelRegression(azimuthD1,
Follins$rotation[domain1Crit], bandwidthD1, numSteps =
                                       1000)
KernRsquaredD1

# Do a permutation test for significance.
rSquaredsD1 <-
    rotKernelRegressionPermutations(azimuthD1,
Follins$rotation[domain1Crit], bandwidthD1, numPerms =
                                        1000)
length(rSquareds)
pD1 <- sum(rSquareds > KernRsquared$rSquared)

#the p-value for this regression.
pD1

#————————————————————————————————————

#Now do the same thing for Domain 3

#DOMAIN 3, n = 32. A series of Geodesic Regressions for every 10° azimuth,
with p-values not calculated.
regressionsD3 <- regressionSweep(Follins, 10, domain3Crit, 0)

# Create a dataframe with the summary information for the regressions
regressionsD3Sum <- data.frame(regressionsD3[[19]])

# Name the columns of the data frame
names(regressionsD3Sum)   =   c("Azimuth",   "Error",   "Min_Eigen",
"R_squared", "Pvalue")

# View the summary table for the output.
regressionsD3Sum

# Plot the R^2 value as a function of Azimuth
```

```
plot(
    as.vector(regressionsD3Sum$Azimuth),
    as.vector(regressionsD3Sum$R_squared),
    xlab = "Azimuth",
    ylab = "R-squared"
)

#————————————————————————————————————————————————————————————

# Define the Azimuth with the highest R-squared value. (In this case,
60)
azimuthD3 <-
    Follins$easting[domain3Crit]     *     sin(60     *     degree)     +
Follins$northing[domain3Crit] *
    cos(60 * degree)

#DOMAIN 3, n = 32. Define the bandwith for a kernal regression. If
desired, uncomment the next line (it may take ~20 minutes to run).
Otherwise, use the pre-calculated value, "bandwidthD3".
bandwidthD3 <-
    rotBandwidthForKernelRegression(Follins$northing[domain3Crit],
Follins$rotation[domain3Crit])
bandwidthD3 <- 0.1331092

#Run the kernel regression
kernelRegD3 <- lapply(
    seq(
        from = min(azimuthD3),
        to = max(azimuthD3),
        by = 100
    ),
    rotKernelRegression,
    azimuthD3,
    Follins$rotation[domain3Crit],
    bandwidthD3,
    numSteps = 1000
)
sapply(kernelRegD3, function(regr)
    regr$error)
sapply(kernelRegD3, function(regr) {
    regr$minEigenvalue > 0
})

# Plot the regression curve
kernelRegCurveD3 <- lapply(kernelRegD3, function(regr)
    regr$r)
oriEqualVolumePlot(Follins$rotation[domain3Crit],
                   curves = list(kernelRegCurveD3),
                   simplePoints = TRUE,
                   oriLineInPlaneGroup)

# Compute the R^2.
KernRsquaredD3 <-
```

```
    rotRsquaredForKernelRegression(azimuthD3,
Follins$rotation[domain3Crit], bandwidthD3, numSteps = 1000)
KernRsquaredD3

# Do a permutation test for significance.
rSquaredsD3 <-
    rotKernelRegressionPermutations(Follins$northing[domain3Crit],
Follins$rotation[domain3Crit], bandwidthD3, numPerms = 1000)
length(rSquareds)
pD3 <- sum(rSquareds > KernRsquaredD3$rSquared)

#the p-value for this regression.
pD3

##============================STATISTICAL
DESCRIPTORS========================================

# 1) Domain 1, n = 23

# Frechet mean (minimizes the Frechet variance).
FrechetMeanDom1 <-
    oriFrechetMean(Follins$rotation[domain1Crit], oriLineInPlaneGroup)
geoStrikeDipRakeDegFromRotation(FrechetMeanDom1)
FrechetVarDom1 <-
    oriVariance(Follins$rotation[domain1Crit],         FrechetMeanDom1,
oriLineInPlaneGroup)

# Plot the FrechetMean in an Equal Angle plot
FrechetCurvesDom1 <-
    lapply(Follins$rotation[domain1Crit], function(r)
        rotGeodesicPoints(FrechetMeanDom1, r, 10))
oriEqualAnglePlot(points  =  Follins$rotation[domain1Crit],  curves  =
FrechetCurvesDom1)

# Print the Strike, Dip, Rake of the Frechet mean.
geoStrikeDipRakeDegFromRotation(FrechetMeanDom1)

# Print the Frechet variance.
FrechetVarDom1

#————————————————————————————————————————————————————————————

# 2) Domain 2, n = 14
#Frechet mean minimizes the Frechet variance.
FrechetMeanDom2 <-
    oriFrechetMean(Follins$rotation[domain2Crit], oriLineInPlaneGroup)
geoStrikeDipRakeDegFromRotation(FrechetMeanDom2)
FrechetVarDom2 <-
    oriVariance(Follins$rotation[domain2Crit],         FrechetMeanDom2,
oriLineInPlaneGroup)

# Plot the FrechetMean in an Equal Angle plot
FrechetCurvesDom2 <-
```

```
    lapply(Follins$rotation[domain2Crit], function(r)
        rotGeodesicPoints(FrechetMeanDom2, r, 10))
oriEqualAnglePlot(points = Follins$rotation[domain2Crit], curves =
FrechetCurvesDom2)

# Print the Strike, Dip, Rake of the Frechet mean.
geoStrikeDipRakeDegFromRotation(FrechetMeanDom2)

# Print the Frechet variance
FrechetVarDom2

#————————————————————————————————————————————————————————————

# 3) Domain 3, n = 32
#Frechet mean minimizes the Frechet variance.
FrechetMeanDom3 <-
    oriFrechetMean(Follins$rotation[domain3Crit], oriLineInPlaneGroup)
geoStrikeDipRakeDegFromRotation(FrechetMeanDom3)
FrechetVarDom3 <-
    oriVariance(Follins$rotation[domain3Crit],         FrechetMeanDom3,
oriLineInPlaneGroup)

# Plot the FrechetMean in an Equal Angle plot
FrechetCurvesDom3 <-
    lapply(Follins$rotation[domain3Crit], function(r)
        rotGeodesicPoints(FrechetMeanDom3, r, 10))
oriEqualAnglePlot(points = Follins$rotation[domain3Crit], curves =
FrechetCurvesDom3)

# Print the Strike, Dip, Rake of the Frechet mean.
geoStrikeDipRakeDegFromRotation(FrechetMeanDom3)

# Print the Frechet variance.
FrechetVarDom3

#————————————————————————————————————————————————————————————

# 4) Dispersion of the data

#The standard way to get at dispersion is to compute the maximum matrix
fisher likelihood. The matrix fisher distribution comprises a kind of
"mean" and a positive definite symmetric matrix, which characterises the
anisotropic dispersion.

# Redefine the rotations to be within one of the symmetric copies
mu <- oriFrechetMean(Follins$rotation, group = oriLineInPlaneGroup)
Follins$rotation <-
    oriNearestRepresentatives(Follins$rotation,     mu,     group     =
oriLineInPlaneGroup)

# Northern Domain, n = 16. Fisher maximum likelihood
mleDom1 <- rotFisherMLE(Follins$rotation[domain1Crit])
mleDom1
```

```
eigen(mleDom1$kHat, symmetric = TRUE, only.value = TRUE)$values

#Central Domain, n = 34. Fisher maximum likelihood
mleDom2 <- rotFisherMLE(Follins$rotation[domain2Crit])
mleDom2
eigen(mleDom2$kHat, symmetric = TRUE, only.value = TRUE)$values

#Southern Domain, n = 79. Fisher maximum likelihood
mleDom3 <- rotFisherMLE(Follins$rotation[domain3Crit])
mleDom3
eigen(mleDom3$kHat, symmetric = TRUE, only.value = TRUE)$values

#===========================INFERENCE====================================
======

# Here, we can perform some statistical techniques to use information
about the samples (each domain) to compute a probability cloud of the
mean of the population(s) from which those samples were taken.
# From numerical work in Davis and Titus (2017), MCMC will work the best
for small sample sizes that have the matrix fisher anisotropy
(eigenvalues) of (large, large, small). All four domains have this
anisotropy, and have too few data points to use the (computationally
quicker) bootstrapping method.

#We'll do both and compare.

# Compute the bootstrap cloud of means
Follins_domain1Boots <-
    oriBootstrapInference(Follins$rotation[domain1Crit],        10000,
oriLineInPlaneGroup)
Follins_domain2Boots <-
    oriBootstrapInference(Follins$rotation[domain2Crit],        10000,
oriLineInPlaneGroup)
Follins_domain3Boots <-
    oriBootstrapInference(Follins$rotation[domain3Crit],        10000,
oriLineInPlaneGroup)

# Compute the Markov chain Monte Carlo cloud of means
Follins_domain1MCMC <-

oricWrappedTrivariateNormalMCMCInference(Follins$rotation[domain1Crit]
,
                                             group                =
oriLineInPlaneGroup,
                                             numCollection = 100)
Follins_domain2MCMC <-

oricWrappedTrivariateNormalMCMCInference(Follins$rotation[domain2Crit]
,
                                             group                =
oriLineInPlaneGroup,
                                             numCollection = 100)
Follins_domain3MCMC <-
```

```
oricWrappedTrivariateNormalMCMCInference(Follins$rotation[domain3Crit]
,
                                             group                    =
oriLineInPlaneGroup,
                                             numCollection = 100)

#————————————————————————————————————————————————————————————

# 1) Northern v. Central domains

# A) Using MCMC

# Construct the 95% confidence ellipsoids from small triangles
tris1_MCMC <-
    rotEllipsoidTriangles(
        Follins_domain1MCMC$mBar,
        Follins_domain1MCMC$leftCovarInv,
        Follins_domain1MCMC$q095,
        numNonAdapt = 4
    )
tris2_MCMC <-
    rotEllipsoidTriangles(
        Follins_domain2MCMC$mBar,
        Follins_domain2MCMC$leftCovarInv,
        Follins_domain2MCMC$q095,
        numNonAdapt = 4
    )

# Plot the MCMC comparison in an equal Volume plot, with 95% confidence
ellipsoids
oriEqualAnglePlot(
    points = c(Follins_domain1MCMC$ms, Follins_domain2MCMC$ms),
    boundaryAlpha = .1,
    axesColors = c("black", "black", "black"),
    fogStyle = "none",
    background = "white",
    triangles = c(tris1_MCMC, tris2_MCMC),
    simplePoints = TRUE,
    colors = c(replicate(length(
        Follins_domain1MCMC$ms
    ), "black"), replicate(
        length(Follins_domain2MCMC$ms), "orange"
    )),
    group = oriTrivialGroup
)

# If you wish to save this figure, maximize the plot window on your
screen before running this line. It will save to your working directory
folder.
afterMaximizingWindow("MCMC_Ahs_d1d2_1.png", "MCMC_WestMt_d1d2_2.png")

# Plot the MCMC comparison in an Equal Area plot
```

```
lineEqualAreaPlotTwo(c(
    lapply(Follins_domain1MCMC$ms, function(s)
        s[1, ]),
    lapply(Follins_domain1MCMC$ms, function(s)
        s[2, ])
),
c(
    lapply(Follins_domain2MCMC$ms, function(s)
        s[1, ]),
    lapply(Follins_domain2MCMC$ms, function(s)
        s[2, ])
))

#.....................................................................

# B) Using bootstrapping

# Construct the 95% confidence ellipsoids from small triangles
tris1_Boot <-
    rotEllipsoidTriangles(
        Follins_domain1Boots$center,
        Follins_domain1Boots$leftCovarInv,
        Follins_domain1Boots$q095,
        numNonAdapt = 4
    )
tris2_Boot <-
    rotEllipsoidTriangles(
        Follins_domain2Boots$center,
        Follins_domain2Boots$leftCovarInv,
        Follins_domain2Boots$q095,
        numNonAdapt = 4
    )

# Plot the bootstrap comparison in an equal Volume plot, with 95%
confidence ellipsoids
rotEqualAnglePlot(
    points = c(
        Follins_domain1Boots$bootstraps,
        Follins_domain2Boots$bootstraps
    ),
    triangles = c(tris1_Boot, tris2_Boot),
    boundaryAlpha = .1,
    axesColors = c("black", "black", "black"),
    fogStyle = "none",
    background = "white",
    simplePoints = TRUE,
    colors = c(replicate(
        length(Follins_domain1Boots$bootstraps), "black"
    ), replicate(
        length(Follins_domain2Boots$bootstraps), "orange"
    ))
)
```

```
# If you wish to save this figure, maximize the plot window on your screen before running this line. It will save to your working directory folder.
afterMaximizingWindow("Boots_Ahs_d1d2_1.png", "Boots_Ahs_d1d2_2.png")

# Plot the bootstrap comparison in an Equal Area plot
lineEqualAreaPlotTwo(c(
    lapply(Follins_domain1Boots$bootstraps, function(s)
        s[1, ]),
    lapply(Follins_domain1Boots$bootstraps, function(s)
        s[2, ])
),
c(
    lapply(Follins_domain2Boots$bootstraps, function(s)
        s[1, ]),
    lapply(Follins_domain2Boots$bootstraps, function(s)
        s[2, ])
))

#————————————————————————————————————————————————————————————

# 2) Northern vs. Southern domains

# A) Using MCMC

# Construct the 95% confidence ellipsoids from small triangles
tris1_MCMC <-
    rotEllipsoidTriangles(
        Follins_domain1MCMC$mBar,
        Follins_domain1MCMC$leftCovarInv,
        Follins_domain1MCMC$q095,
        numNonAdapt = 5
    )
tris3_MCMC <-
    rotEllipsoidTriangles(
        Follins_domain3MCMC$mBar,
        Follins_domain3MCMC$leftCovarInv,
        Follins_domain3MCMC$q095,
        numNonAdapt = 5
    )

# Plot the bootstrap comparison in an equal Volume plot, with 95% confidence ellipsoids
oriEqualAnglePlot(
    points = c(Follins_domain1MCMC$ms, Follins_domain3MCMC$ms),
    triangles = c(tris1_MCMC, tris3_MCMC),
    boundaryAlpha = 0.1,
    axesColors = c("black", "black", "black"),
    fogStyle = "none",
    background = "white",
    simplePoints = TRUE,
    colors = c(replicate(length(
        Follins_domain1MCMC$ms
```

```
    ), "black"), replicate(length(
        Follins_domain3MCMC$ms
    ), "blue")),
    group = oriTrivialGroup
)

# If you wish to save this figure, maximize the plot window on your
screen before running this line. It will save to your working directory
folder.
afterMaximizingWindow("MCMC_Ahs_d1d3_1.png", "MCMC_Ahsd1d3_2.png")

# Plot the MCMC comparison in an Equal Area plot
lineEqualAreaPlotTwo(c(
    lapply(Follins_domain1MCMC$ms, function(s)
        s[1, ]),
    lapply(Follins_domain1MCMC$ms, function(s)
        s[2, ])
),
c(
    lapply(Follins_domain3MCMC$ms, function(s)
        s[1, ]),
    lapply(Follins_domain3MCMC$ms, function(s)
        s[2, ])
))

#.............................................................

# B) Using bootstrapping
# Construct the 95% confidence ellipsoids from small triangles
tris1_Boot <-
    rotEllipsoidTriangles(
        Follins_domain1Boots$center,
        Follins_domain1Boots$leftCovarInv,
        Follins_domain1Boots$q095,
        numNonAdapt = 4
    )
tris3_Boot <-
    rotEllipsoidTriangles(
        Follins_domain3Boots$center,
        Follins_domain3Boots$leftCovarInv,
        Follins_domain3Boots$q095,
        numNonAdapt = 4
    )

# Plot the bootstrap comparison in an equal Volume plot, with 95%
confidence ellipsoids.
rotEqualAnglePlot(
    points = c(
        Follins_domain1Boots$bootstraps,
        Follins_domain3Boots$bootstraps
    ),
    triangles = c(tris1_Boot, tris3_Boot),
    boundaryAlpha = .1,
```

```
    axesColors = c("black", "black", "black"),
    fogStyle = "none",
    background = "white",
    simplePoints = TRUE,
    colors = c(replicate(
        length(Follins_domain1Boots$bootstraps), "black"
    ),
    replicate(
        length(Follins_domain3Boots$bootstraps), "blue"
    ))
)

# If you wish to save this figure, maximize the plot window on your screen before running this line. It will save to your working directory folder.
afterMaximizingWindow("Boots_Ahs_d1d3_1.png", "Boots_Ahs_d1d3_2.png")

# Plot the bootstrap comparison in an Equal Area plot
lineEqualAreaPlotTwo(c(
    lapply(Follins_domain1Boots$bootstraps, function(s)
        s[1, ]),
    lapply(Follins_domain1Boots$bootstraps, function(s)
        s[2, ])
),
c(
    lapply(Follins_domain3Boots$bootstraps, function(s)
        s[1, ]),
    lapply(Follins_domain3Boots$bootstraps, function(s)
        s[2, ])
))



#————————————————————————————————————————

# 3) Central vs. Southern domains

# A) Using MCMC

# Construct the 95% confidence ellipsoids from small triangles
tris2_MCMC <-
    rotEllipsoidTriangles(
        Follins_domain2MCMC$mBar,
        Follins_domain2MCMC$leftCovarInv,
        Follins_domain2MCMC$q095,
        numNonAdapt = 4
    )
tris3_MCMC <-
    rotEllipsoidTriangles(
        Follins_domain3MCMC$mBar,
        Follins_domain3MCMC$leftCovarInv,
        Follins_domain3MCMC$q095,
        numNonAdapt = 4
```

```
    )

# Plot the bootstrap comparison in an equal Volume plot, with 95%
confidence ellipsoids
oriEqualAnglePlot(
    points = c(Follins_domain2MCMC$ms, Follins_domain3MCMC$ms),
    triangles = c(tris2_MCMC, tris3_MCMC),
    boundaryAlpha = 0.1,
    axesColors = c("black", "black", "black"),
    fogStyle = "none",
    background = "white",
    simplePoints = TRUE,
    colors = c(replicate(
        length(Follins_domain2MCMC$ms), "orange"
    ), replicate(length(
        Follins_domain3MCMC$ms
    ), "blue")),
    group = oriTrivialGroup
)

# If you wish to save this figure, maximize the plot window on your
screen before running this line. It will save to your working directory
folder.
afterMaximizingWindow("MCMC_Ahs_d2d3_1.png", "MCMC_Ahs_d2d3_2.png")

# Plot the MCMC comparison in an Equal Area plot
lineEqualAreaPlotTwo(c(
    lapply(Follins_domain2MCMC$ms, function(s)
        s[1, ]),
    lapply(Follins_domain2MCMC$ms, function(s)
        s[2, ])
),
c(
    lapply(Follins_domain3MCMC$ms, function(s)
        s[1, ]),
    lapply(Follins_domain3MCMC$ms, function(s)
        s[2, ])
))

#....................................................................


# B) Using bootstrapping

# Construct the 95% confidence ellipsoids from small triangles
tris2_Boot <-
    rotEllipsoidTriangles(
        Follins_domain2Boots$center,
        Follins_domain2Boots$leftCovarInv,
        Follins_domain2Boots$q095,
        numNonAdapt = 4
    )
tris3_Boot <-
```

```
    rotEllipsoidTriangles(
        Follins_domain3Boots$center,
        Follins_domain3Boots$leftCovarInv,
        Follins_domain3Boots$q095,
        numNonAdapt = 4
    )

# Plot the bootstrap comparison in an equal Volume plot, with 95% confidence ellipsoids
oriEqualAnglePlot(
    points = c(
        Follins_domain2Boots$bootstraps,
        Follins_domain3Boots$bootstraps
    ),
    triangles = c(tris2_Boot, tris3_Boot),
    boundaryAlpha = 0.1,
    axesColors = c("black", "black", "black"),
    fogStyle = "none",
    background = "white",
    simplePoints = TRUE,
    colors = c(replicate(
        length(Follins_domain1Boots$bootstraps), "orange"
    ), replicate(
        length(Follins_domain3Boots$bootstraps), "blue"
    )),
    group = oriTrivialGroup
)

# If you wish to save this figure, maximize the plot window on your screen before running this line. It will save to your working directory folder.
afterMaximizingWindow("Boots_Ahs_d2d3_1.png", "Boots_Ahs_d2d3_2.png")

# Plot the bootstrap comparison in an Equal Area plot
lineEqualAreaPlotTwo(c(
    lapply(Follins_domain2Boots$bootstraps, function(s)
        s[1, ]),
    lapply(Follins_domain2Boots$bootstraps, function(s)
        s[2, ])
),
c(
    lapply(Follins_domain3Boots$bootstraps, function(s)
        s[1, ]),
    lapply(Follins_domain3Boots$bootstraps, function(s)
        s[2, ])
))
```