dsld: A Socially Relevant Tool for Teaching Statistics

Taha Abdullah
Department of Computer Science, University of California, Davis
and
Arjun Ashok
Department of Computer Science, University of California, Davis
and
Brandon Estrada
Department of Computer Science, University of California, Davis
and
Norman Matloff
Department of Computer Science, University of California, Davis
and
Aditya Mittal
Department of Statistics, University of California, Davis

February 10, 2025

Abstract

The growing power of data science can play a crucial role in addressing social discrimination, necessitating nuanced understanding and effective mitigation strategies for biases. “Data Science Looks At Discrimination” (**dsld**) is an R and Python package designed to provide users with a comprehensive toolkit of statistical and graphical methods for assessing possible discrimination related to protected groups such as race, gender, and age. The package addresses critical issues by identifying and mitigating confounders and reducing bias against protected groups in prediction algorithms.

In educational settings, **dsld** offers instructors powerful tools to teach statistical principles through motivating real-world examples of discrimination analysis. The inclusion of an 80-page Quarto book further supports users—from statistics educators to legal professionals—in effectively applying these analytical tools to real-world scenarios.

Keywords: Data science; Fair machine learning; Statistics education; Discrimination analysis; Confounder analysis; Quarto notebook

1 Introduction

Statistics—the class students love to hate! It's hard to think of a course less popular, yet required by more majors, than statistics. Recent studies have found a negative student perception of statistical courses, ranging from undergraduate to graduate level course work (Naidu and Arumugam, 2014; Dani and Al Quraan, 2023). Perhaps relabeling as "data science" will help a bit; however, the subject is badly in need of better motivation. To aid this, a number of remedies have been proposed, ranging from the flipped classroom (Kovacs *et al.*, 2021) to stories involving Disney characters (Peters, 2019). The American Statistical Association also has suggestions (Carver *et al.*, 2016).

Our package, `dsld` (Data Science Looks at Discrimination), takes a different approach by appealing to students' awareness of social issues (Bowen *et al.*, 2017). The software provides both analytical and graphical/tabular methods for investigating possible bias related to race, gender, age, and other potential sensitive variables. Specifically, two broad categories are addressed:

- **Detection of discrimination:** This section focuses on identifying and compensating for confounding variables. For instance, is there a gender gap in wages after taking into account confounders such as age, occupation, number of weeks worked, etc.?
- **Addressing bias in prediction:** This section focuses on the reduction of bias in the application of predictive algorithms. For example, consider a tool to aid in granting loan applications. If an applicant's race is included as a predictor – either explicitly or through proxy variables – how can one mitigate its effect?

The value of the package is greatly enhanced via the use of a tightly integrated open source textbook, written in Quarto (Allaire *et al.*, 2024). The book is not a user manual for the package, but instead is a detailed treatment of the statistical concepts themselves, illustrated with `dsld` examples.

This is no toy. On the contrary, the package will be quite useful in social science research, internal HR analyses, and in discrimination litigation. Both parametric and nonparametric regression models are available.

We note other R packages focusing on analysis of discrimination and related issues: `divseg` (Kenny, 2022) is concerned with urban racial segregation; `genderstat` (Arafin Ayon, 2023) is "...an exhaustive tool developed for the R...programming environment, explicitly devised to expedite quantitative evaluations in the field of gender studies;" `segregation` (Elbers, 2021) is a tool for the calculation of relationships in two-way contingency tables, including with grouping, with a typical intended use case being analysis of urban racial segregation. Several packages address the issue of fairness in prediction, including `fairML` (Scutari, 2023a); `fairmodels` (Wiśniewski and Biecek, 2021); and `fairness` (Kozodoi and V. Varga, 2021).

The organization of the remainder of this paper is as follows: Section 2 introduces the package and Quarto book. Section 3 covers detection of discrimination, and Section 4 addresses reduction of bias in prediction. Finally, the paper concludes with a discussion and future perspectives in Section 5.

Some notation: We have a response variable Y related to a vector of covariates X, and a sensitive variable S; the latter may be continuous, binary or categorical. Y can be continuous or binary, with coding 1 and 0 in the latter case. In predicting Y in a new case, the predicted value is denoted by $\widehat{Y}$.

2 The dsld Package and Quarto Book

The `dsld` package was developed in 2023 by seven undergraduates at the University of California, Davis, working under the direction of Norman Matloff. It currently consists of 24 functions, plus an associated open source textbook. It is available on CRAN, and the latest version is always in `https://github.com/matloff/dsld`.

2.1 Relation to qeML package

As noted, the package consists of both analytical and graphical/tabular routines. Many of the former functions are wrappers for routines in other packages, adding discrimination-specifc functionality.

In the context of educational usage of `dsld`, simplicity of the interface is of high importance, in order to be accessible by a broad student audience. It attains this in many cases via its use of the `qeML` package. The latter is designed to provide an exceptionally simple, direct user interface. For instance, to fit a linear model to predict **mpg** in R's built-in `mtcars` dataset, in `qeML` one makes the simple call:

```
qeLin(data = mtcars, yName = ‘mpg’)
```

No preparatory calls, no setup of any kind, just an easy request for a fit. To do the same, but with random forests, just call:

```
qeRF(data = mtcars, yName = ‘mpg’)
```

Use `qeKNN(data = mtcars, yName = ‘mpg’)` for K-Nearest Neighbors and so on. Flexibility is retained via the number of default arguments.

2.2 Introducing dsld

As noted, many `dsld` functions wrap functions from other packages while adding material specific to discrimination analysis. To make this concrete, consider `dsldLinear`, which wraps `qeLin` (which in turn wraps R's `lm`). Here are the call forms of the two functions:

```
qeLin(data, yName, noBeta0 = FALSE, holdout = floor(min(1000,
    0.1 * nrow(data))))
dsldLinear(data, yName, sName, interactions = FALSE,
    sComparisonPts = NULL, useSandwich = FALSE)
```

where `yName` is the name of the response variable Y.

We see that `dsldLinear` has some new arguments not present in `qeML`, which are central to the discrimination analysis:

- `sName`: the name of the sensitive variable S.
- `interactions`: if true, include interactions terms with S.
- `sComparisonPts`: an argument related to interactions, explained further below.
- `useSandwich`: if true, indicates usage of the sandwich method to deal with heteroscedasity (Boe *et al.*, 2023). Not directly related to discrimination analysis.

If `interactions` is set, interactions between the sensitive variable and the predictors/features in our dataset will be modeled. Actually, separate linear models will be fit for each level of the sensitive variable, which is essentially statistically equivalent. The presence or absence of interaction terms plays an important role in the package and in the Quarto book. Is this a subtantial difference in mean Y, given the covariates X, across the levels of the R factor S? If so, is this difference uniform with respect to X?

In the case of interactions there is no single "treatment effect" of the sensitive variable. We can no longer speak of "the" difference in mean wage between men and women, as this difference may vary by, say, age. Accordingly, the user can specify several points at which to compare the effects of the different levels of S; estimated differences in mean conditional Y are given for the user-specified points `sComparisonPts`.

The package includes numerous graphical/tabular functions, some of them wrappers to existing packages such as `freqparcoord` and others standalone. These are used both for preliminary exploration of one's data, and also for visual illustrations of the results found analytically.

For example, one might use `dsldLogit` to estimate parameters in a logistic model predicting passage of the bar examination, based on scores on the Law School Admission Test, and then supplement the results with graphics such as Figure 1. There seem to be substantial differences among the races for low LSAT values, but not as much at the higher end.

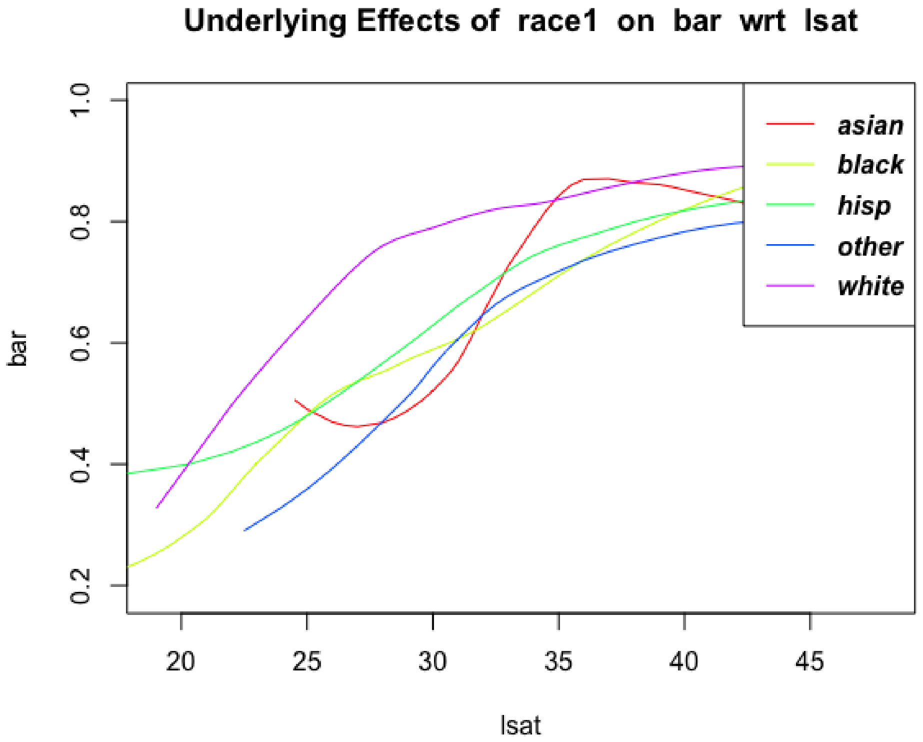

Figure 1: Predicted probability of passing the bar exam by LSAT scores highlighting racial disparities among different groups. Student with GPA below 2.70.

Just as all `qeML` argument lists begin with `data` and `yName`, the same is true with `dsld`, with a standard third argument being `sName`.

2.3 The Role of Nonparametric Regression Models

As noted in the last section, realistic analyses of discrimination must in many cases account for interaction effects between the covariates X and the sensitive variable S. Often, one must also consider nonlinear effects.

One may address nonlinearity via low-degree polynomial models, but modern computing power is such that a laptop computer can easily handle fully nonparametric regression methods, even in fairly large datasets. In `dsld`, we feature Random Forests (RFs) as our primary non-parametric approach (Hastie *et al.*, 2009).

Note that the statistical term *non-parametric regression models* corresponds in the computer science community with *machine learning* (ML), though there is arguably a difference in interpretation, a running debate ever since statistician Leo Breiman's famous essay, "Statistical Modeling: The Two Cultures," was published (Breiman, 2001b).

We chose RFs for a couple of reasons. First, it is arguably the most familiar nonparametric regression method among statisticians, having been developed primarily in the mid-90s and 2000s by statisticians (Breiman, 2001a). (Less well known, but also significant, is the work by computer scientists such as Tin Ho Kam and her coauthors (Ho, 1995).) Second, RFs can be easily explained to nonstatisticians, due to their "if this, then that" flow chart nature.

2.4 The Quarto book

The Quarto book, about 100 pages in length, is the product of all eight of the package authors. As mentioned, though it uses `dsld` examples, it presents the statistical concepts, rather than serving as a user manual for the package. Moreover, it emphasizes understanding of the methods beyond just their definition, in practical senses. Below is an excerpt:

> One may have specific confounders in mind for a particular analysis, but it is often unclear as to which to use, or for that matter, why not use them all?...
>
> Technically, almost any variable is a confounder. The impact may quite minuscule, but through a long chain of relations among many variables, there will usually be at least some connection, though again possibly very faint...
>
> ...there are several issues to consider not using the full set of variables i.e. every variable other than Y and S:
>
> - It may result in overfitting, resulting in large standard errors.
> - It is unwieldy, difficult to interpret. Many treatments of these issues speak of a desire for a "parsimonious" model.

The math involved in the book is minimized, and the material should be accessible to students who have taken a (noncalculus-based) course in elementary statistics. (There are a few optional "starred sections" covering advanced topics.)

2.5 Datasets

Both through the `dsld` package itself, and through the packages that it wraps, a number of built-in datasets are available to greatly enhance its usefulness as a teaching tool. Note that these datasets tend to be of an observational nature, an issue covered in the `Quarto book`.

3 Detecting Discrimination

Discrimination is a significant social issue in the United States and many other countries. There is lots of available data with which one might investigate possible discrimination. But how might such investigations be conducted? There is a rich array of classical parametric methods for this, and recently nonparametric regression methods are in increasing use, such as in HR management (Frissen *et al.*, 2022). The `dsld` package offers both graphical and analytical tools to detect potential biases, which are particularly beneficial for students. These tools can enhance understanding and intuition during classroom discussions by connecting them to broader social contexts.

In this section, we will use the `lsa` dataset on law school admissions, focusing on *race* as the sensitive variable. The racial categories included are Asian, Black, White, Hispanic, and Other. This dataset is available through the `dsld` package for further exploration and analysis.

3.1 Graphical/Tabular Methods

Effective graphs and visualizations significantly enhance students' understanding of data. The `dsld` package offers various graphical methods to investigate discrimination issues, including `dsldDensityByS`, `dsldConditDisparity`, `dsldConfounders` and `dsldScatterplot3D`.

Consider investigating potential discrimination in the context of college and graduate school admissions. There has been growing criticism of standardized testing, arguing it disproportionately favors students with more family income and other resources (Foreiter, 2021). Indeed, many studies have suggested unfair discrepancies in test results between Black and White students (Dixon-Roman *et al.*, 2013). As a result, many institutions have removed standardized testing requirements like the SAT and GRE from their application processes. This debate underscores the importance of examining biases and motivates our study of the `lsa` dataset.

dsldConditDisparity

Suppose we wish to analyze the effect of LSAT scores on passing the bar examination to investigate potential racial disparities. We may calculate differences in the probability of passing the bar for a given LSAT score among different racial groups. The function `dsldConditDisparity` is particularly suited for this purpose, as it visualizes conditional disparities, providing insights into how LSAT scores and bar examination outcomes might vary between racial groups.

```
dsldConditDisparity(data = lsa, yName = 'bar', sName = 'race1',
    xName = 'lsat', condits = NULL)
```

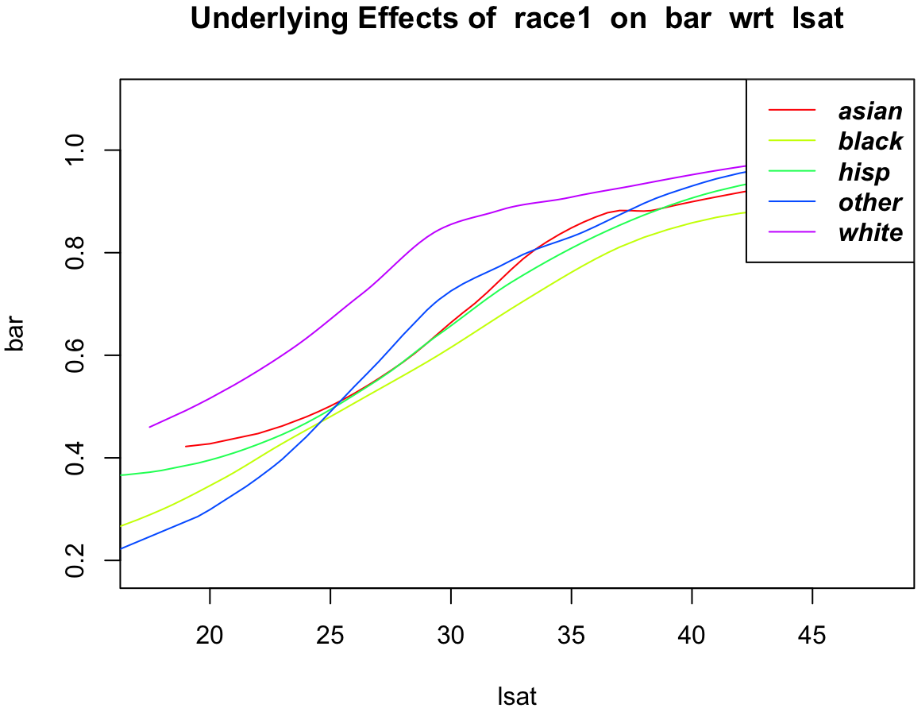

Figure 2: Estimated probability of passing the bar exam by LSAT scores, highlighting racial disparities among different groups. All students.

Notably, all non-White groups initially exhibit similar outcomes in Figure 2, with higher passing probabilities observed for White students up through mid-range LSAT levels. However, as mentioned earlier, all racial groups display comparable passing probabilities at higher LSAT ranges. This observation suggests the potential relevance of racial interaction terms in formal analyses and serves as a valuable starting point for classroom discussions on disparities in educational outcomes based on race.

The function also includes optional argument `condits` ("conditions"). For instance, we could restrict the domain of students to those with lower undergraduate GPAs (≤ 2.70) as shown in Figure 1 and conduct similar analyses.

dsldDensityByS

To further explore the relationship between race and LSAT scores, we can generate a density plot using the `dsldDensityByS` function. In general, this method enables users to visualize densities of a response variable Y segmented by a sensitive variable S, with bandwidth control. This visualization provides insight into potential differences of the distribution of LSAT scores across different racial groups.

This is complementary to our previous graph, Figure 2, which indicated that Whites pass the bar exam at higher rates than do non-Whites, even at the same LSAT levels. That difference may be exacerbated if Whites also tend to do better on the LSAT. Let's see:

```
dsldDensityByS(data = lsa, cName = 'lsat', sName = 'race1')
```

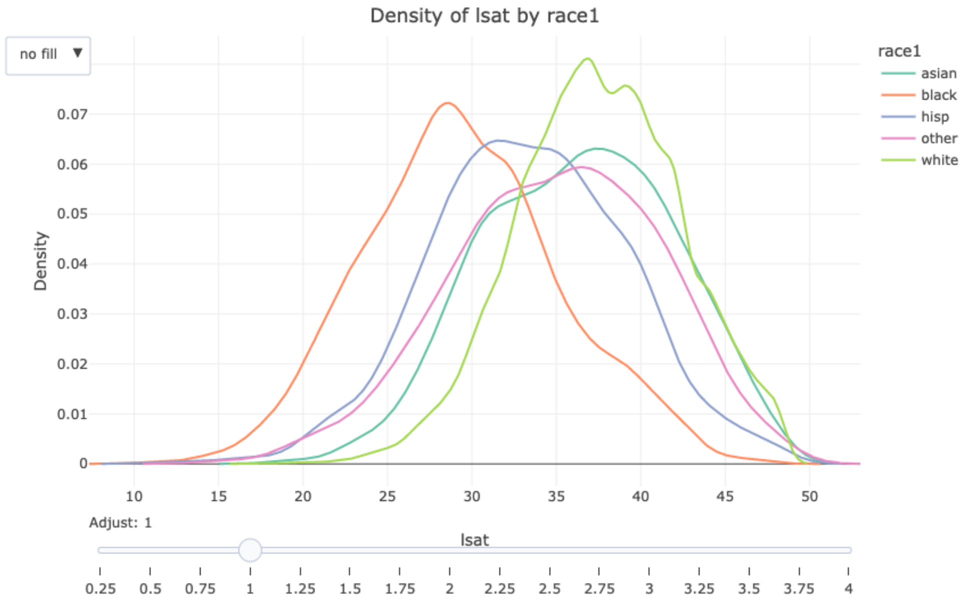

Figure 3: Density plot of LSAT scores segmented by race. The bandwidth parameter can be thought as analogous to controlling the bin width in a histogram to adjust the figure granularity. Default bandwidth is set at 1.

The function `dsldConfounders` acts similarly, but it is Plotly-based, and thus features user interactive access.

The density plot above illustrates the distribution of LSAT scores across different racial groups, highlighting a potential bias. However, that apparent effect may be influenced by confounding variables. Confounders influence both dependent and independent variables, possibly leading to spurious associations (McNamee, 2003).

dsldScatterPlot3D

For example, analyzing relationships among LSAT scores, undergraduate GPA, and family income can reveal potential confounding interactions. We can visualize these relationships segmented by race using `dsldScatterPlot3D` using a 3D perspective, offering insights into correlations and disparities across racial groups with respect to family income and GPA.

```
dsldScatterPlot3D(data = lsa, yNames = c('lsat', 'fam_inc',
    'ugpa'), sName = 'race1', pointSize = 4)
```

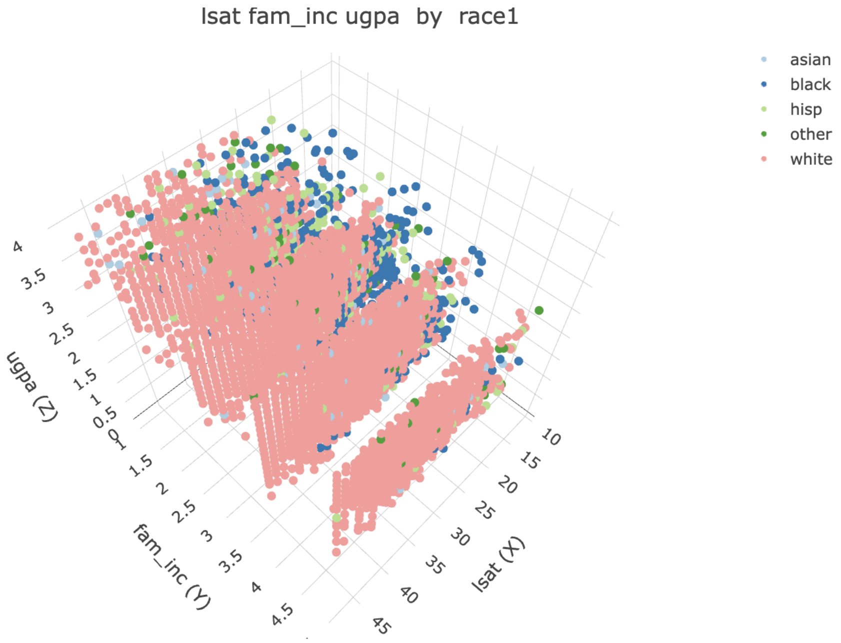

Figure 4: 3-dimensional scatterplot showing family income, race, and gender. Lower family income is mostly Black and Latino students; higher income levels are predominantly White. Lower LSAT scores are mostly non-white across all income levels. UGPA trends similarly to LSAT but less strongly.

While this analysis is merely exploratory, Figure 4 suggests that family income may *not* be the major confounder in the relationship between race and LSAT scores that some had assumed. Non-White students appear to consistently show lower scores across various income levels, prompting further discussion on how race influences exam outcomes while in consideration of other variables. Note that the output from `dsldScatterPlot3D` can be rotated in R interactive mode for better visualization.

dsldFreqParCoord

In addition to `dsldScatterPlot3D`, `dsld` provides another function to aid in visualizing data in more than two dimensions. The `dsldFreqParCoord` function plots such interactions through *parallel coordinates* (Inselberg, 2008). Consider a dataset with p columns, where each column corresponds to a vertical axis in the graph. For every data row, a polygonal line is drawn horizontally, with its height at vertical section i representing the value of variable for that row. Typically, each variable is centered and scaled. Each row of data generates a distinctive pattern. The function `dsldFreqParCoord` facilitates the visualization of the m most frequently occurring patterns from each level of the sensitive variable S. Two patterns are considered equivalent if they are proximate in a k-nearest neighbor sense, with k being an argument passed by the user. The following call form can help create the plot as displayed in Figure 5 below.

```
### use a subset of lsa: ‘fam_inc’,‘ugpa’,‘gender’,‘lsat’,‘race1’
lsa1 <- lsa[,c(‘fam_inc’,‘ugpa’,‘gender’,‘lsat’,‘race1’)]
dsldFreqPCoord(data = lsa1, m = 75, sName = ‘race1’)
```

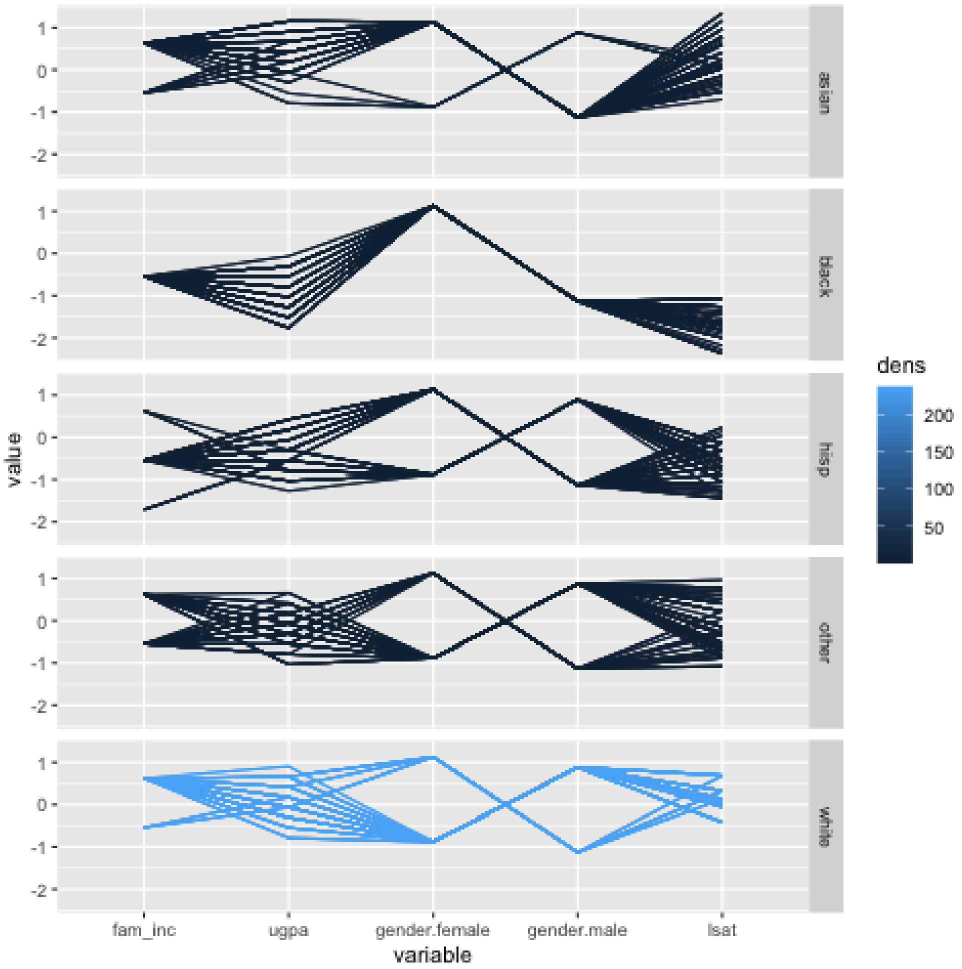

Figure 5: Visualization of the most frequently occurring patterns across different racial groups using parallel coordinates.

Several notable patterns can be seen. Black students exhibit the least variation, with the most common pattern characterized by female students from slightly below-average income families, low undergraduate grades, and significantly low LSAT scores. Patterns for Whites form a mirror inversion of those of Black students, though with greater gender diversity. Asian students display trends similar to Whites. Hispanic students, however, demonstrate a wide range of income levels and grades, yet consistently low LSAT scores.

The graphical analysis tools provided by `dsld` are invaluable for educators and students in academic settings, aiding in the visualization of complex statistical relationships. For instance, examining LSAT score distributions among different racial groups can initiate discussions on fairness and bias in standardized testing. Similarly, analyzing how family income influences academic outcomes encourages exploration of socioeconomic disparities. These visual methods foster critical thinking by enabling students to interpret real-world data and understand the implications of statistical findings in societal contexts. The Quarto notebook offers a comprehensive analysis of these results and includes relevant insights from other datasets as well.

dsldCHunting

Users can utilize methods offered by the `dsld` package to identify potential confounding variables in their own analyses. For instance, the `dsldCHunting` function employs random forests to evaluate the importance of predictors in predicting Y (without S) and S (without Y). These predictors are identified as "important predictors" of Y and S and aid in the detection of potential confounding factors. For each i from 1 to `intersectDepth`, the function reports the intersection of the top i important predictors of Y and S. This helps identify potential confounders, with larger values of i indicating inclusion of progressively weaker predictors.

```
dsldCHunting(data = lsa, yName = ‘bar’, sName = ‘race1’,
    intersectDepth = 10)
```

	decile3	lsat	cluster	ugpa	decile1	age	fulltime	gender	fam_inc
impForY	0.121	0.063	0.007	0.007	0.006	0.003	0.002	0.001	-0.001
impForS	0.029	0.033	0.023	0.012	0.025	0.003	0.000	0.002	0.006

Table 1: Importance measures for variables in “impForY” and “impForS”. These values indicate feature importance in prediction of Y (without S) and S (without Y), higher values indicate stronger predictors.

In the context of the `lsa` dataset, importance measures provide insights into predictors influencing the probability of passing the bar, which may act as confounders concerning race. Table 1 above shows the importance measures: the first row indicates predictors for the probability of passing the bar from other parameters (excluding race), while the second row shows predictors for race from other variables (excluding bar passage). For instance, variables that predict Y (probability of passing the bar) well include decile3, LSAT, and cluster. In predicting S (race), the top predictors are again LSAT, decile3, and cluster. By focusing on variables correlated with both Y and S, we can identify potential confounders. For example, the intersection of the top three predictors for Y and S includes decile3, LSAT, and cluster. The analyst has flexibility in choosing the number of confounding variables, ranging from just the top two to more, depending on the individual needs and objectives. Furthermore, results from `dsldConfounders` can further add to one’s analysis. In particular, this method provides a density plot similar to `dsldDensityByS` for continuous predictors. For categorical variables, it displays the frequency of the sensitive variable occurring at each level of that variable. The Quarto book offers additional insights of bias detection using various datasets and examples.

3.2 Analytical

While graphical approaches serve as initial steps in exploratory data analysis, this should be followed up with formal methods. the `dsld` package also provides analytical methods for formal investigation of potential instances of discrimination. Continuing with the LSAT example, suppose we wish to examine pairwise differences in estimated mean LSAT scores across racial groups using a linear model. In this section, we’ll see how `dsldLinear` facilitates this.

A key issue is whether to include in our model interactions between our covariates X and the sensitive variable S, a major point of the `dsld` package. In educational settings in particular, it is important that students understand that in some cases, there may not be such thing as “the” difference between one group and another; the difference may vary considerably with X. We saw this in Figure 2 above, in which Y was an indicator variable for passage of the bar examination.

Here we fit a linear model with Y set to LSAT score, with an eye toward investigating possible bias in the test. Let’s compare the non-interactive and interactive cases:

- The **no-interaction** case uses the following call:

  ```
  z1 = dsldLinear(data = lsa, yName = 'lsat', sName = 'race1',
      interactions = FALSE)
  summary(z1)
  ```

- Here is the **with-interaction** call:

  ```
  newData = lsa[c(1,10,100,1000,10000),]
  z2 = dsldLinear(data = lsa, yName = 'lsat', sName = 'race1',
      interactions = TRUE, sComparisonPts = newData)
  summary(z2)
  ```

The key point is that not only does the second call differ from the first in the value of the `interactions` argument, but also in that the second call includes the argument `sComparisonPts`. What is happening here? In addition to calculating the usual estimated regression coefficients and so on, `dsldLinear`, the function facilitates comparison across levels of S, as follows.

As noted earlier, in a no-interactions model, we can speak of, for instance, the Black and White difference in mean LSAT scores, independent of the values of X. But if interactions of X and S are assumed, there is no notion of “the” Black vs. White difference. That difference will vary according to the values of X, so the function estimates the difference in mean Y at values of X that are of interest to the user, as specified in `sComparisonPts`.

In general, a linear model with interaction terms involving a categorical variable with m levels amounts to fitting m separate linear models.

Thus, in the with-interactions model in the LSAT example, `dsldLinear` essentially fits five distinct linear models – one for each race.

Let's take a look at the output of the two models. First, the no-interactions case:

```
$`Summary Coefficients`
     Covariate    Estimate StandardError       PValue
1  (Intercept) 31.98578856   0.448435264 0.000000e+00
2          age  0.02082458   0.005841758 3.641634e-04
3      decile1  0.12754812   0.020946536 1.134602e-09
4      decile3  0.21495015   0.020918737 0.000000e+00
5      fam_inc  0.30085804   0.035953051 0.000000e+00
6         ugpa -0.27817274   0.080430542 5.430993e-04
7   gendermale  0.51377385   0.060037102 0.000000e+00
8   race1black -4.74826307   0.198088318 0.000000e+00
9    race1hisp -2.00145969   0.203504412 0.000000e+00
10 race1other  -0.86803104   0.262528590 9.449471e-04
11 race1white   1.24708760   0.154627086 6.661338e-16
...
$`Sensitive Factor Level Comparisons`
  Factors Compared Estimates Standard Errors      P-Value
1    asian - black  4.748263       0.1980883 0.000000e+00
2     asian - hisp  2.001460       0.2035044 0.000000e+00
3    asian - other  0.868031       0.2625286 9.449471e-04
4    asian - white -1.247088       0.1546271 6.661338e-16
5     black - hisp -2.746803       0.1863750 0.000000e+00
6    black - other -3.880232       0.2515488 0.000000e+00
7    black - white -5.995351       0.1409991 0.000000e+00
8     hisp - other -1.133429       0.2562971 9.764506e-06
9     hisp - white -3.248547       0.1457509 0.000000e+00
```

```
10      other - white  -2.115119          0.2194472  0.000000e+00
```

We see for example that the estimated difference in dummy variables for Black and White scores is 6.0000 points with a standard error of 0.1410. In view of the fact that LSAT scores in this dataset ranged from 11 to 48 points, a 6-point difference seems to be very concerning.

On the other hand, the effect of family income is rather small, and indeed gives instructors a chance for a "teachable moment," as follows. A key concept often emphasized by instructors of statistics courses is the difference between *statistical significance* versus *practical significance*. The results here form a perfect example. Consider the variable for family income, which is measured in quintiles, with values 1,2,3,4,5. The coefficient for family income is "very highly significant," with a p-value of 0 to six figures, yet the effect size is small, about 0.3. The difference between, say, the 3rd and 4th quintiles in terms of mean LSAT is about 0.3 point. That's minuscule in relation to the general LSAT scores — as mentioned, ranging from 11 to 48 points here — and the Black-White difference of 6 points.

3.3 Causal Models

The function `dsldMatchedATE`, wrapping `Matching::Match` offers matched-pairs analysis (Huber, 2023). Direct pairing can be done, or one may opt to use propensity scores, using either `glm` or `qeML::KNN` for a nonparametric k-Nearest Neighbors approach. Let's estimate the "treatment effect" of being female in a possible gender wage gap. We use the `svcensus` data, which includes information on individuals' age, education, occupation, income, weeks worked.

```
data(svcensus)
summary(dsldMatchedATE(svcensus,'wageinc','gender','male'))
```

```
# Estimate...   9634.5
# SE.........   380.03
# T-stat.....   25.352
# p.val......  < 2.22e-16
```

```
#
# Original number of observations..............   20090
# Original number of treated obs...............   15182
# Matched number of observations...............   20090
# Matched number of observations  (unweighted).   20090
```

Men are estimated to earn $9634.50 more than similar women, with a standard error of $380.03.

Using either logit or k-NN to predict gender, the estimated wage gap becomes:

```
summary(dsldMatchedATE(svcensus,'wageinc','gender','male',
   propensFtn='glm'))
#
# Estimate...   10332
# SE.........   408.39
# ...

summary(dsldMatchedATE(svcensus,'wageinc','gender','male',
   propensFtn='knn',k=50))


# Estimate...   9877.8
# SE.........   439.73
# ...
```

The function `dsldIamb` deals with *causal discovery*. Typically a causal Directed Acyclic Graph (DAG) is merely a reflection of the analyst's "gut feeling" about relations among the variables. The DAG cannot be derived from the data itself without making some very stringent assumptions (Shalizi, 2024; Scutari, 2023a).

The `bnlearn` package includes various models of this sort, and `dsldIamb` wraps the `iamb` function, with which a DAG may be constructed from the data.

```
data(svcensus)
# iamb does not accept integer data
svcensus$wkswrkd <- as.numeric(svcensus$wkswrkd)
```

```
svcensus$wageinc <- as.numeric(svcensus$wageinc)
iambOut <- dsldIamb(svcensus)
plot(iambOut)
```

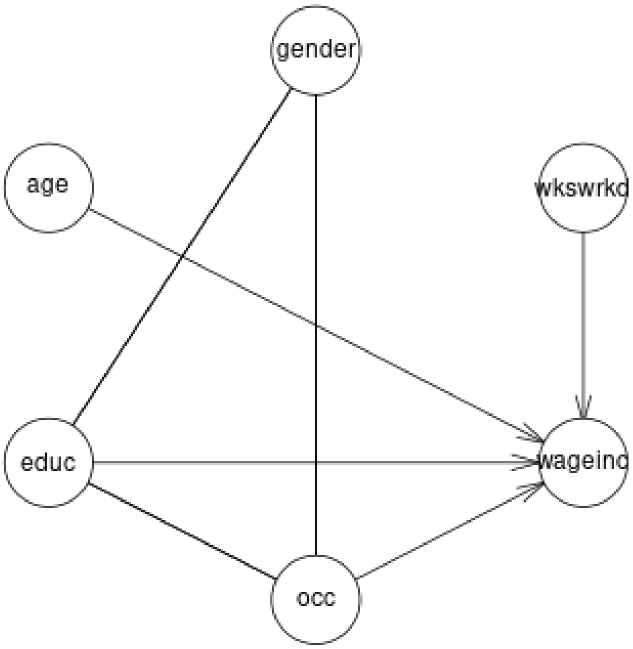

Figure 6: DAG generated under iamb assumptions

Figure 6 shows the result. Interestingly, the only variable that is *not* modeled as causal for wage income is gender, though gender is shown as having a weak, non-directional relation with occupation and gender.

Again, the graph is computed under very restrictive assumptions, and the function should be regarded as exploratory.

4 Addressing Bias in Predictive Algorithms

In predicting Y from X, we may wish to avoid the influence of S. In evaluating an application for a mortgage, say, we hope that our assessment does not discriminate against women or minorities. Such issues have been the subject of intensive research in recent years. There is even a conference devoted to the effort, the ACM Conference on Fairness, Accountability, and Transparency (ACM FAccT). Here, we present some of `dsld`'s capabilities in this regard.

The "Hello World" example for this realm of research involves the COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) algorithm developed by Northpointe, a commercial entity. The algorithm was designed to predict a defendant's likelihood of recidivism, thus aiding judges in determining sentences. Note that, as a commercial product, the algorithm is a "black-box," of completely unknown details.

The algorithm has faced criticism following an investigation by the publication *ProPublica*, which suggested that the algorithm exhibited bias against Black defendants (Angwin *et al.*, 2016). Northpointe has contested these findings, arguing that *ProPublica*'s analysis is flawed. *ProPublica* issued a rejoinder, but in any case, the debate over COMPAS underscores the critical need to address fairness.

Beyond the legal system, predictive methods have rapidly expanded into various commercial applications in recent years, including cybersecurity, healthcare, and e-commerce (Sarker, 2021). These models now play a pivotal role in decision-making processes with significant consumer impact. This rise has highlighted concerns about algorithmic fairness in predictive methods (Wehner and Köchling, 2020; Chen, 2023).

Broadly speaking, "fairness" in this context refers to minimizing the influence of sensitive attributes (e.g., race, gender, religion, etc.) on an algorithm's predictions (Oneto and Chiappa, 2020).

There are several main issues:

- **Defining unfairness:** Several measures have been proposed.
- **Reducing unfairness:** For a given algorithm, how can we ameliorate its unfairness while maintaining an acceptable predictive accuracy level?

- **Dealing with proxies:** Typically there will be proxy variables which, through correlation with S, can result in the latter influencing $\widehat{Y}$ even if S is omitted from the analysis entirely.

The dsld package addresses the need for fair prediction by providing a convenient and intuitive interface to a range of fairness-constrained modeling approaches. In this section, we will use the COMPAS and Law Schools datasets to the showcase tradeoff between fairness and accuracy on both regression and classification settings. For the COMPAS dataset, we will predict the probability of recidivism using race as the sensitive feature. For the Law Schools dataset, we will predict LSAT scores with race as the sensitive variable. Both datasets are accessible via the dsld package for further exploration and analysis.

4.1 Relevant Methods Provided by dsld

The dsld package provides wrappers for several functions from the faiML package (Scutari, 2023a) and for the Explicitly Deweighted Features (EDF) methods developed in Matloff and Zhang (2022). Let's consider the fairML functions first.

Komiyama *et al.* (2018), Zafar *et al.* (2017) and Scutari (2023a) take ridge-regression-like approaches to reducing the influence of S on the predicted value, $\widehat{Y}$. All these methods are implemented in the fairML package (Scutari, 2023b), with dsld providing interfaces.

Marco Scutari's work in the fairML package offers a statistical approach to fair machine learning, building on previous efforts by imposing a ridge-regression-like penalty on the coefficient(s) of S. For example, let's fit a Fair Generalized Ridge Regression model — a fair adaptation of the Generalized Linear Models, and make a prediction of recidivism in the COMPAS dataset. As an example case, take someone like observation 188 in our dataset — age 23, no prior convictions, male and Black.

```
cps <-
   compas1[,c(‘age’,‘priors_count’,‘sex’,‘two_year_recid’,‘race’)]
z <- dsldFgrrm(data=cps, yName=‘two_year_recid’, sName=‘race’,
   unfairness=0.1)
newx <- cps[188,-4]
predict(z,newx)
```

```
# 0.508936
```

The `unfairness` argument must be a value in (0,1], with values near 0 being nearly perfectly fair. This controls the Fairness-Utility Tradeoff.

Here the algorithm reports an estimated probability of recidivism of about 51%. What if we had placed no restriction on the role of race?

```
z <- dsldFgrrm(data=cps, yName=‘two_year_recid’, sName=‘race’,
    unfairness=1.0)
predict(z,newx)
# 0.5723716
```

Now the probability of recidivism jumps to 57%.

The `dsld` package also includes wrappers for functions from the work on Explicitly Deweighted Features (Matloff and Zhang, 2022). This approach omits S entirely and mitigates the impact of each proxy variable through user-specified hyperparameters. The latter enable the user to select a desired point on the Fairness-Utility Tradeoff spectrum. The package offers implementations of linear/logistic regression (works like ridge regression, except with different “λ” values for each proxy variable), random forests (different proxy variables have different probabilities of being used for splitting a node), and k-nearest neighbors (k-NN) (Euclidean distance but with different weights for each proxy variable).

Let’s apply k-NN on the same example as above.

```
z <- dsldQeFairKNN(data=cps, yName=’two_year_recid’,
   sName=‘race’, deweightPars= list(age=0.5,priors_count=0.1),
   yesYVal=‘Yes’)
predict(z,newx)
# 0.52
```

Here, we entirely omit the race variable from our dataset and deweight the proxies ‘age’ and ‘priors count’ by 0.5 and 0.2 (a weight of 1.0 means no deweighting) to reduce their predictive power.

4.2 Assessing Fairness and Utility

One possible measure of fairness is to require that $\widehat{Y}$ and S are statistically independent (Johndrow and Lum, 2019). However, but this eliminates any use of proxies and thus the ability to achieve a Fairness-Utility Tradeoff.

A broader view would be to ask that $\widehat{Y}$ and S have low correlation. See for instance Li *et al.* (2023), Kozodoi *et al.* (2022), Deho *et al.* (2022), Mary *et al.* (2019), and Baharlouei *et al.* (2019) apply work from Lee *et al.* (2022) and Roh *et al.* (2023).

The `dsld` package continues this theme:

- **Measuring Accuracy:** To assess utility, we utilize a holdout set to compute test accuracy from predictions generated by our algorithms. In regression contexts, accuracy is defined as mean absolute prediction error (MAPE), while for classification settings, we use misclassification rate.

- **Measuring Fairness:** The measure of fairness is the correlation between $\widehat{Y}$ and S, computed using the Kendall Tau statistic. This choice is preferred over the more commonly used Pearson correlation due to its flexibility with both continuous and categorical inputs. While Pearson is geared towards continuous numeric variables, Kendall Tau is also usable for binary or ordinal integer-valued variables.

4.3 Proxy Hunting

A proxy is a variable that indirectly represents a protected attribute, potentially introducing bias in decision-making even when the protected attribute is not explicitly used. For example, race or religion may be linked to a city or neighborhood in a city (Morse *et al.*, 2020).

To detect potential proxies in a dataset, the `dsld` package provides methods such as `dsldOHunting` for users to make informed decisions. Specifically, this function calculates the Kendall Tau correlation between the sensitive feature and all other features in the dataset. Consider use of `dsldOHunting` on the COMPAS example presented earlier:

```
dsldOHunting(data = cps, yName =  ‘two_year_recid’, sName = ‘race’)
                                   age    priors_count  sex.Female  sex.Male
```

```
# race.African-American  -0.156   0.175        -0.0414    0.041
# race.Asian              0.016  -0.030        -0.0216    0.021
# race.Caucasian          0.147  -0.100         0.0686   -0.068
# race.Hispanic           0.022  -0.068        -0.0261    0.026
# race.Native American    0.002   0.023        -0.0065    0.006
# race.Other              0.002  -0.087        -0.0129    0.012
```

This suggests, say, that age and prior convictions count might be treated as proxies for being African-American, and possibly deweighting them if one uses EDF methods.

4.4 Measuring Fairness versus Utility

In this section, we will showcase the application of `dsld` functions on the fairness-utility trade-off in both regression and classification settings using `dsldFrrm` and `dsldZlrm`, respectively. Users can also apply a similar method for other functions to facilitating efficient grid search across various parameters. Note: To minimize sampling variation in our results, we will incorporate 25 holdout sets by setting the `reps` argument to 25.

We analyze the fairness-utility trade-off in regression tasks by varying the unfairness parameter on the `Law School Admissions` dataset, where the goal is to predict LSAT while considering race as the sensitive attribute. In this example, we will use `dsldFrrm`, specifically testing the following values of the unfairness parameter: 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, and 0.99. The following code shows how to implement and plot the grid-search results:

```
# Grid search for fairness vs. utility
ex1 <- dsldUtilsFairness(law.school.admissions, ‘lsat’, ‘race1’,
   ‘dsldFrrm’, unfairness = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75,
   0.8, 0.9, 0.99), yesYVal = NULL, reps = 10)
print(ex1)

 d      MAPE     asian      black      hisp       other       white
0.01  3.63402 0.0103227  0.0557155 0.0394955 0.00979468  0.0668236
0.05  3.55174 0.0399833  0.1344877 0.0680349 0.02208106  0.1533411
```

```
0.10  3.51030  0.0407389   0.1762693  0.0985790  0.02496056   0.1987062
0.25  3.42200  0.0602939   0.2680973  0.1509890  0.05899232   0.3095009
0.50  3.38134  0.0446075   0.3079764  0.1688408  0.06858589   0.3401874
0.75  3.38159  0.0086641   0.3107674  0.1729236  0.06133157   0.3231249
0.80  3.38485  0.0481784   0.3042744  0.1737998  0.07459950   0.3449968
0.90  3.38457  0.0519573   0.3121371  0.1775541  0.06527029   0.3508078
0.99  3.38704  0.0344068   0.3049084  0.1876693  0.06376829   0.3422318
```

The results above provide the mean absolute prediction error and the correlations for each racial group. The trends indicate that as fairness is prioritized by lowering correlations, accuracy tends to decline. The fairness versus utility can also be plotted as follows:

```
plot(exp1)
```

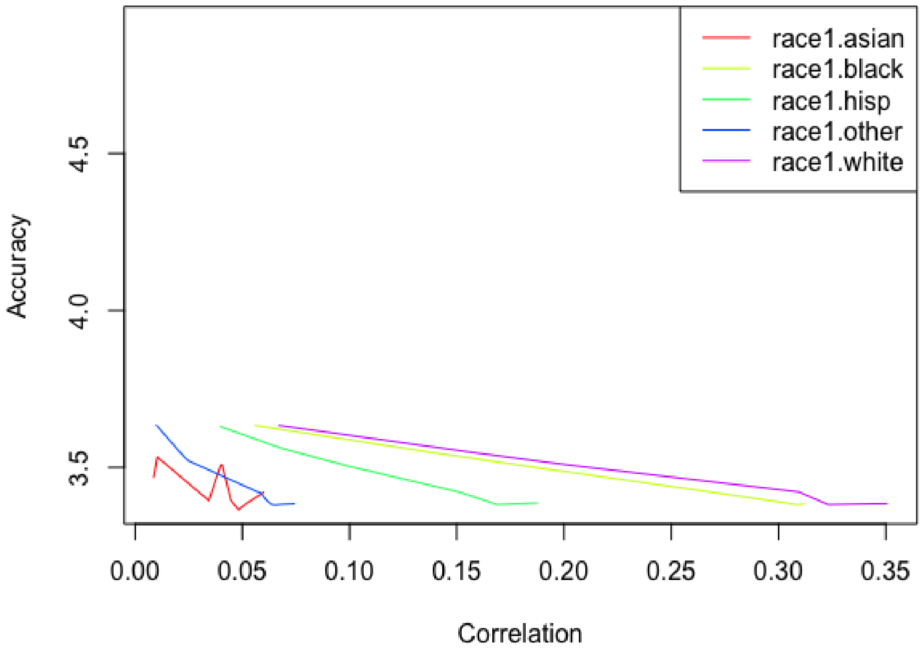

Figure 7: Fairness versus accuracy trade-off with the unfairness parameter treated as a hidden variable. Regression case using the Law School Admissions data, where the Y-axis is MAPE and the X-axis is correlation between predicted LSAT score and race.

Figure 7 the fairness versus accuracy results for various values of the unfairness parameter as a hidden variable. The results indicate that smaller values of the unfairness parameter result in minimal correlation between the predicted LSAT scores and race, which increases as the unfairness parameter rises. The mean absolute prediction error exhibits a decreasing trend, underscoring the trade-off between fairness and utility as the unfairness parameter changes. The variation in correlation across different values is most pronounced for both whites and African Americans, demonstrating the effects on fairness on these subgroups.

Now, let's consider the example of `dsldZlrm` using the `COMPAS` dataset for classification. We are predicting the probability of recidivism with race as the sensitive variable. For simplicity, we have pre-processed the data to focus on three race categories: Caucasian, African-Americans, and Hispanics.

```
# Grid search for fairness vs. utility
ex3 <- dsldUtilsFairness(compas1, ‘two_year_recid’, ‘race’,
   ‘dsldZlrm’,  unfairness=c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75,
   0.8, 0.9, 0.99), yesYVal = 'Yes', reps = 25, omitS = FALSE)
print(ex3)


 d          Accuracy    African-American    Caucasian       Hispanic
0.01        0.3402587     0.04783121       0.03715734      0.02044614
0.05        0.3012126     0.09942695       0.06808573      0.05814433
0.10        0.2871059     0.12965339       0.09115132      0.07130585
0.25        0.2783347     0.19502954       0.15669713      0.07498811
0.50        0.2747373     0.20122967       0.16165470      0.07690962
0.75        0.2719078     0.19437225       0.15177063      0.08187104
0.80        0.2735247     0.19297602       0.15103262      0.08086585
0.90        0.2710590     0.18738216       0.14435984      0.08237444
0.99        0.2762328     0.19350622       0.15492846      0.07510151
```

The plot method can be called again to display the fairness versus utility graph across various values of the unfairness parameter.

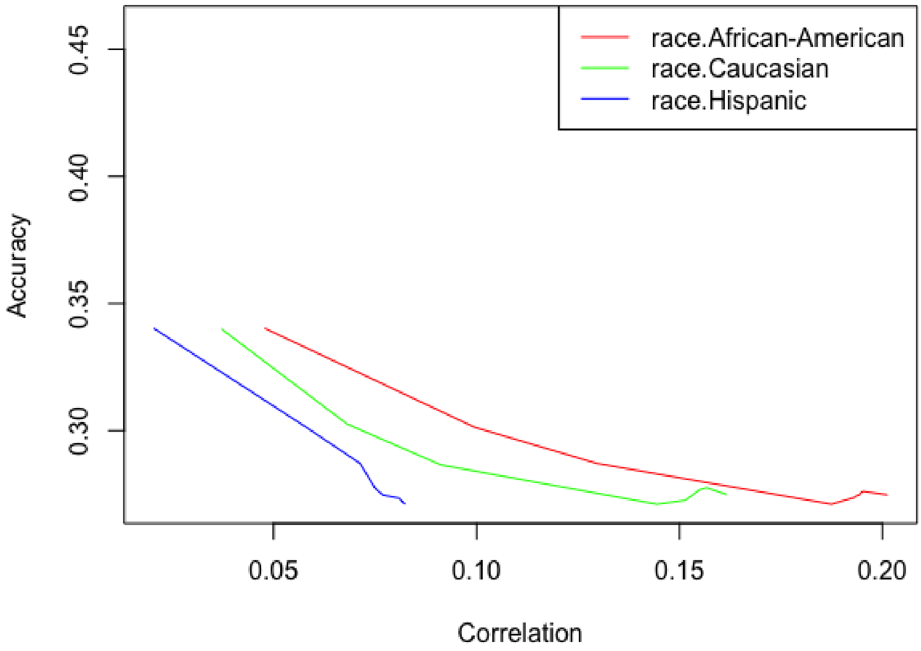

Figure 8: Fairness versus accuracy trade-off with the unfairness parameter treated as a hidden variable. Classification case using the COMPAS data, where the Y-axis is misclassification rate and the X-axis is correlation between predicted recidivism and race.

Similar patterns can be observed in the classification case, with misclassification rate decreasing as the correlation between S and the predicted Y increases. Changes in the unfairness parameter can significantly impact correlations, especially for African Americans, who exhibited the most substantial range of changes. A similar analysis can be conducted using other `EDF-FAIR` and `fairML` wrappers. For the `EDF-FAIR` wrappers, replace the range of the `unfairness` parameter with various values of the `DeweightPars` argument (which also allows for deweighting multiple combinations of proxy variables if several are present).

This section aims to enhance student understanding by illustrating the fairness-utility trade-off. The diverse functions offered by `dsld` provide a broad range of fair modeling techniques, enabling a deeper exploration of the balance between fairness and accuracy in prediction.

5 Discussion

In this paper, we present "Data Science Looks at Discrimination" (`dsld`) as a powerful tool for educating students in statistics and related fields on the concepts of fair machine learning. The software includes a variety of analytical and graphical tools that help students intuitively explore and visualize potential sources of bias and discrimination. The fair machine learning wrappers encapsulate several fairness-constrained machine learning models, allowing for the quick and seamless deployment of bias mitigation algorithms. Additionally, the accompanying Quarto notebook provides a comprehensive foundation, offering the high-level statistical background necessary to understand fair machine learning in an engaging and accessible manner, requiring only a basic understanding of mathematics.

As machine learning becomes an increasingly important tool for statistical analysis and prediction, it is crucial for users to recognize the risks of its misapplication. The `dsld` package not only bridges the gap between traditional and fairness-constrained machine learning algorithms but also serves as an effective educational resource. It equips both current and future students with the knowledge and tools to responsibly apply practical machine learning systems while balancing fairness with utility and effort.

SUPPLEMENTARY MATERIAL

The package can be accessed here: https://github.com/matloff/dsld/tree/master. All relevant datasets, functions, and implementation details can be found through the Github.

References

Allaire, J., Teague, C., Scheidegger, C., Xie, Y., and Dervieux, C. (2024). Quarto.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016). Machine bias.

Arafin Ayon, S. M. M. (2023). *genderstat: Quantitative Analysis Tools for Gender Studies*. R package version 0.1.3.

Baharlouei, S., Nouiehed, M., Beirami, A., and Razaviyayn, M. (2019). R\'enyi fair inference. *arXiv preprint arXiv:1906.12005*.

Boe, L. A., Lumley, T., and Shaw, P. A. (2023). Practical considerations for sandwich variance estimation in two-stage regression settings. *American Journal of Epidemiology*.

Bowen, G., Gordon, N., and Chojnacki, M. (2017). Advocacy through social media: Exploring student engagement in addressing social issues. *Journal of Higher Education Outreach and Engagement*, **21**, 5–30.

Breiman, L. (2001a). Random forests. *Machine Learning*, **45**, 5–32.

Breiman, L. (2001b). Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical Science*, **16**(3), 199 – 231.

Carver, R. H. F., Everson, M., Gabrosek, J., Horton, N. J., Lock, R. H., Mocko, M., Rossman, A., Roswell, G., Velleman, P. F., Witmer, J. A., and Wood, B. L. (2016). Guidelines for assessment and instruction in statistics education (gaise) college report 2016.

Chen, Z. (2023). Ethics and discrimination in artificial intelligence-enabled recruitment practices. *Humanities and Social Sciences Communications*, **10**.

Dani, A. and Al Quraan, E. (2023). Investigating research students' perceptions about statistics and its impact on their choice of research approach. *Heliyon*, **9**(10), e20423.

Deho, O., Zhan, C., Li, J., Liu, J., Liu, L., and le, T. (2022). How do the existing fairness metrics and unfairness mitigation algorithms contribute to ethical learning analytics? *British Journal of Educational Technology*, **53**, 1–22.

Dixon-Roman, E., Everson, H., and Mcardle, J. (2013). Race, poverty and sat scores: Modeling the influences of family income on black and white high school students' sat performance. *Teachers College Record*, **115**.

Elbers, B. (2021). A method for studying differences in segregation across time and space. *Sociological Methods & Research*, **52**(1), 5–42.

Foreiter, K. (2021). How to get away with socioeconomically discriminating against low income law school applicants: Wealth masking as merit.

Frissen, R., Adebayo, K., and Nanda, R. (2022). A machine learning approach to recognize bias and discrimination in job advertisements. *AI & SOCIETY*, **38**, 1–14.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *Random Forests*, pages 1–18.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

Huber, M. (2023). *Causal Analysis: Impact Evaluation and Causal Machine Learning with Applications in R*. MIT Press.

Inselberg, A. (2008). *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Advanced series in agricultural sciences. Springer New York.

Johndrow, J. E. and Lum, K. (2019). An algorithm for removing sensitive information: Application to race-independent recidivism prediction. *The Annals of Applied Statistics*, **13**(1), 189 – 220.

Kenny, C. T. (2022). divseg: Compute diversity and segregation indices.

Komiyama, J., Takeda, A., Honda, J., and Shimao, H. (2018). Nonconvex optimization for regression with fairness constraints. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2737–2746. PMLR.

Kovacs, P., Kuruczleki, E., Kazar, K., Liptak, L., and Racz, T. (2021). Modern teaching methods in action in statistical classes. *Statistical Journal of the IAOS*, pages 899–919.

Kozodoi, N. and V. Varga, T. (2021). *fairness: Algorithmic Fairness Metrics*. R package version 1.2.2.

Kozodoi, N., Jacob, J., and Lessmann, S. (2022). Fairness in credit scoring: Assessment, implementation and profit implications. *European Journal of Operational Research*.

Lee, J., Bu, Y., Sattigeri, P., Panda, R., Wornell, G., Karlinsky, L., and Feris, R. (2022). A maximal correlation approach to imposing fairness in machine learning. In *ICASSP*

2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3523–3527. IEEE.

Li, Y., Chen, H., Xu, S., Ge, Y., Tan, J., Liu, S., and Zhang, Y. (2023). Fairness in recommendation: Foundations, methods and applications.

Mary, J., Calauzènes, C., and Karoui, N. E. (2019). Fairness-aware learning for continuous attributes and treatments. In *International Conference on Machine Learning*.

Matloff, N. and Zhang, W. (2022). A novel regularization approach to fair ml. *arXiv preprint arXiv:2208.06557*.

McNamee, R. (2003). Confounding and confounders. *Occupational and environmental medicine*, **60**, 227–34; quiz 164, 234.

Morse, L., Kane, G., and Awwad, Y. (2020). Protected attributes and "fairness through unawareness".

Naidu, R. and Arumugam (2014). Non-statistics major student's attitude towards introductory statistics course at public universities.

Oneto, L. and Chiappa, S. (2020). Fairness in machine learning. In *Recent trends in learning from data: Tutorials from the inns big data and deep learning conference (innsbddl2019)*, pages 155–196. Springer.

Peters, J. (2019). Playing with stats: Ideas for incorporating fun into the teaching of statistics.

Roh, Y., Lee, K., Whang, S. E., and Suh, C. (2023). Improving fair training under correlation shifts. In *International Conference on Machine Learning*, pages 29179–29209. PMLR.

Sarker, I. (2021). Machine learning: Algorithms, real-world applications and research directions.

Scutari, M. (2023a). fairml: A statistician's take on fair machine learning modelling. *arXiv preprint arXiv:2305.02009*.

Scutari, M. (2023b). *fairml: Fair Models in Machine Learning*. R package version 0.8.

Shalizi, C. (2024). *Advanced Data Analysis from an Elementary Point of View*.

Wehner, M. and Köchling, A. (2020). Discriminated by an algorithm: A systematic review of discrimination and fairness by algorithmic decision-making in the context of hr recruitment and hr development. *BuR - Business Research*, pages 1–54.

Wiśniewski, J. and Biecek, P. (2021). fairmodels: A flexible tool for bias detection, visualization, and mitigation. *arXiv preprint arXiv:2104.00507*.

Zafar, M. B., Valera, I., Rogriguez, M. G., and Gummadi, K. P. (2017). Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*, pages 962–970. PMLR.