PharmaSUG 2024 - Paper RW-275

Win a PS5! How to Run and Compare Propensity Score Matching Performance Across Multiple Algorithms in Five Minutes or Less

Catherine Briggs, Sherrine Eid, Samiul Haque, Robert Collins, SAS Institute, Inc.

ABSTRACT

With the increased use of real-world data (RWD) in clinical and healthcare settings, having a comprehensive comparison of propensity score matching (PSM) algorithms available to researchers is vital. In this paper, we will discuss different avenues for PSM and show the strengths and weaknesses of each using simulated data. We will cover aspects of performance of the algorithms from statistical measures to computing resources. The final part of the paper will demonstrate the effect of the matching algorithms on estimating the causal effect of treatment on an outcome. This paper will use matching algorithms from SAS, R, and Python and show their results through a SAS Viya Visual Analytics Dashboard.

INTRODUCTION

Propensity score matching offers an exciting opportunity to look at casual inference without the burden of conducting a randomized control trial. You can address challenges of confounding variables and use robust models to determine treatment effects from observational health data. The statistical process for PSM was originally published in 1983 by Rosenbaum & Rubin and many programming languages now have algorithms to run the method with little statistical coding required from the researcher. We will describe the steps of running a PSM analysis and show examples in SAS®, R, and Python. Each language possesses strengths for different research objectives and resource constraints. Diverse tools are necessary given the circumstance of resources, talent, and data. This outline of multiple paths for PSM offers insights into the use of the analytic technique with real-world data and provide a guide for implementation that best suits the needs of each study. We show how SAS Viya Visual Analytics can seamlessly run models from these languages and display consistent results for interpretation. We use a synthetic data set of one-thousand subjects in a long-term prospective fibromyalgia study to showcase implementation, outputs, and results using of each of these languages.

WHAT IS PROPENSITY SCORE MATCHING?

Propensity score matching is used most widely in RWD to analyze observational data to obtain unbiased causal treatment effect estimates. The ‘golden standard’ of causal effect studies is the randomized control trial (RCT). In an RCT, subjects who meet study criteria are randomized to treatment or control arms, to minimize the chance their prior characteristics have a bearing on the treatment. This removes confounding variables biasing the causal relationship between treatment and outcome. The purpose of using a propensity score (PS) in observational studies is to create the balance in distributions of the baseline confounders between interventions, so that estimating the causal treatment effect is like a RCT. Once conditioned on the propensity score, each subject has the same chance of receiving treatment. In this way, PSM mimics randomization when randomization isn’t possible. These scores are used to match the treated and untreated (control) subjects for a more comparable and balanced study population than using an entire observational study group.

For example, if you study the effect of knee surgery on knee pain three years after initial symptoms, you need a population that has indications of knee pain with some who have surgery and some who do not. Many personal factors can determine who receives surgery including age, physical fitness level, body mass index (BMI), health insurance status, etc. Some of those factors influence knee pain directly, like BMI and physical fitness level. To avoid the bias of BMI or other characteristics unduly influencing final pain level, matching subjects on these means that covariate can’t inflate (or deflate) the true value of knee surgery to reduce pain. Ultimately, in this case and others, it’s unethical to have a placebo surgery treatment arm. Observational data is the best option to learn about casual effects of surgery on pain and PSM removes biases of variables confounding treatment and outcome. It can be used in any situation where measured covariates influence both the given treatment and the measured outcome.

This method to isolates the casual effect of a treatment or intervention from the many factors that confound the relationship between the treatment and outcome. It requires consideration of the trade-off between bias reduction and the potential sample size loss during matching depending on the strictness of matched pair requirements. The PS also removes the nuance of interactions and some interpretability from the analysis since subjects who have similar scores can have vastly different covariate measures. There is always the chance for additional residual confounding and biased results due to unmeasured covariates or biased PS modeling. With proper reporting of methods and statistical rigor, PSM provides a wealth of insights from previously untapped data.

CODING FOR PROPENSITY SCORE MATCHING

Propensity Score Matching has three main parts: (1) Calculating the propensity score for all subjects, (2) matching observations from the treatment and control groups, and (3) assessing the effectiveness of matching on balancing measured covariates. You then compare the outcome between the two groups. If all the covariates are statistically equivalent, any differences in the outcome can be attributed to the intervention or unmeasured covariates. Some procedures calculate the PS and matching pairs in one call (PROC PSMACTH and MatchIt) and others require multiple steps (Python). In the former case, there is an option to skip the PS modeling and provide previously calculated scores for matching. With any analytic coding, labeling output datasets and including clear reporting of model features is essential, especially when investigating multiple matching methods before testing for casual effect. Below are three code snippets for each of the computing languages. These are the simplest invocation using the defaults for all aspects of PSM for each algorithm, some coded and some not. We recommend coding these default options in practice to avoid confusion for other researchers less knowledgeable about the algorithms.

The data for this demonstration is a ‘plasmode’ (Gadbury et al. 2008, Franklin et al. 2014) version of The Real World Examination of Fibromyalgia: Longitudinal Evaluation of Cost and Treatments (REFLECTIONS) conducted by Robinson et al. (2012). The PSM modeling tests the effect of opioid vs non-opioid pain medication on end of study pain scores. Many factors contribute to the type of pharmaceutical intervention and you want to avoid those factors biasing the effect of treatment on final pain scores of subjects. While these data are representative of authentic RWD the results in this paper may not apply to data of differing size, complexity, and objective. Data cleaning and wrangling before and after PSM is not shown. Furthermore, results should not be used for any medical or personal purpose.

SAS: PROC PSMATCH

SAS (v. Viya 2023.12) has a single procedure to calculate the propensity score, complete matching, assess balance, and output indicated plots and data. PROC PSMATCH offers easy changes for matching methods, ratios, caliper width and more nuanced features of PSM. The Standardized Mean Differences plot gives you a quick and clear way to see the effect of the matching method on the data. For this example, one might choose to match for Gender exactly, given the deviation from 0 of the matched observations, as a next step example in a PSM analysis. This procedure currently doesn’t support models for creating the propensity score other than logistic regression within the procedure. Using PROC HPSPLIT, can provide scores from a decision tree that can be easily used for matching by replacing the ‘psmodel’ statement with ‘psdata’ and indicating the propensity score column name. Below is the code used for this paper:

```
proc psmatch data = casuser.REFL3 region=cs(extend=0);
   class cohort Gender Race Dr_Rheum Dr_PrimCare;
   psmodel cohort(Treated="opioid")= Gender Race Age BMI_B BPIInterf_B
         BPIPain_B CPFQ_B FIQ_B GAD7_B ISIX_B PHQ8_B PhysicalSymp_B SDS_B
         Dr_Rheum Dr_PrimCare;
   match method=OPTIMAL(k=1) stat=lps caliper= . ;
   assess lps var=(Gender Race Age BMI_B BPIInterf_B  BPIPain_B CPFQ_B
         FIQ_B GAD7_B ISIX_B PHQ8_B PhysicalSymp_B SDS_B Dr_Rheum
         Dr_PrimCare);
   output out=matched_data;
run;
```

Figure 1

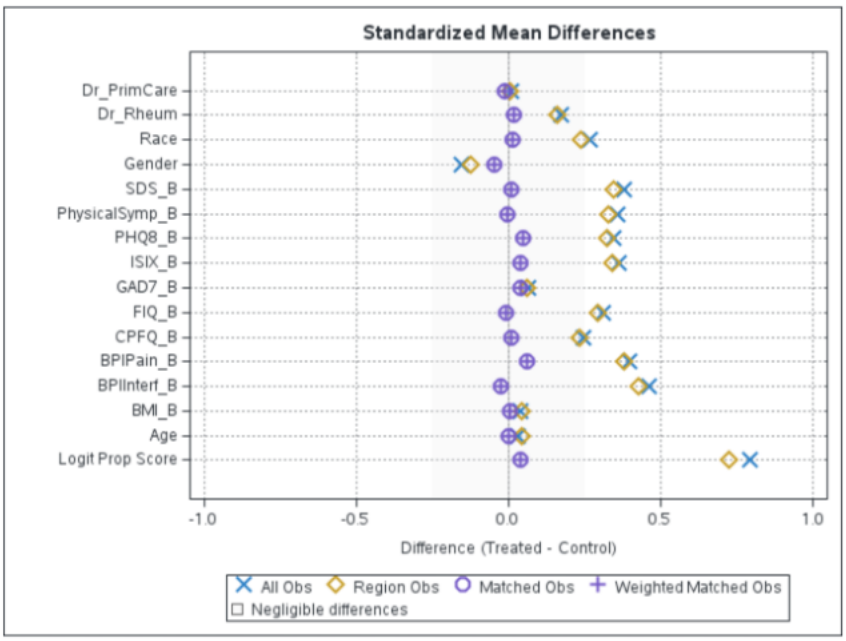


Figure 1: SAS Standardized Mean Differences Plot from PROC PSMATCH.

R: MATCHIT

Propensity score matching in R (v: 4.3.0) is well documented and adaptable. In R, there are multiple libraries for 'out of the box' PSM analysis. For this guide, we chose to use MatchIt (v: 4.5.5) for its wealth of matching options, longevity as a package, and multiple data-driven examples. These features are the strengths of using MatchIt. It is trivial to change matching methods and you can easily replicate examples online to learn different features of the package. One clear advantage is that using advanced machine learning (ML) models for the propensity score calculation is as simple as changing a function value from "glm" to "randomforest", for example. Finding results, observations propensity scores, and plotting can take some time for those unfamiliar with the additional functions required to produce them. Here, Age has a larger standardized difference in the matched subjects vs. unmatched. This could be addressed by setting a stricter caliper width to matching on the Age variable since exactly matching on age might severely reduce the number of matched pairs. Below is the code used for this paper:

```
results <- matchit(cohort~Gender+Race+Age+BMI_B+BPIInterf_B+BPIPain_B
          +CPFQ_B+FIQ_B+GAD7_B+ISIX_B+PHQ8_B+PhysicalSymp_B+SDS_B+Dr_Rheum+
          Dr_PrimCare,
   data = refl3,
   distance = "glm",
   link = "logit",
   method = "optimal",
   ratio = 1,
   caliper = NULL)
summary(results, un=TRUE)
plot(summary(results, un=TRUE))
matched_data<-match.data(results, include.s.weights = TRUE,
          drop.unmatched = FALSE)
```

Figure 2

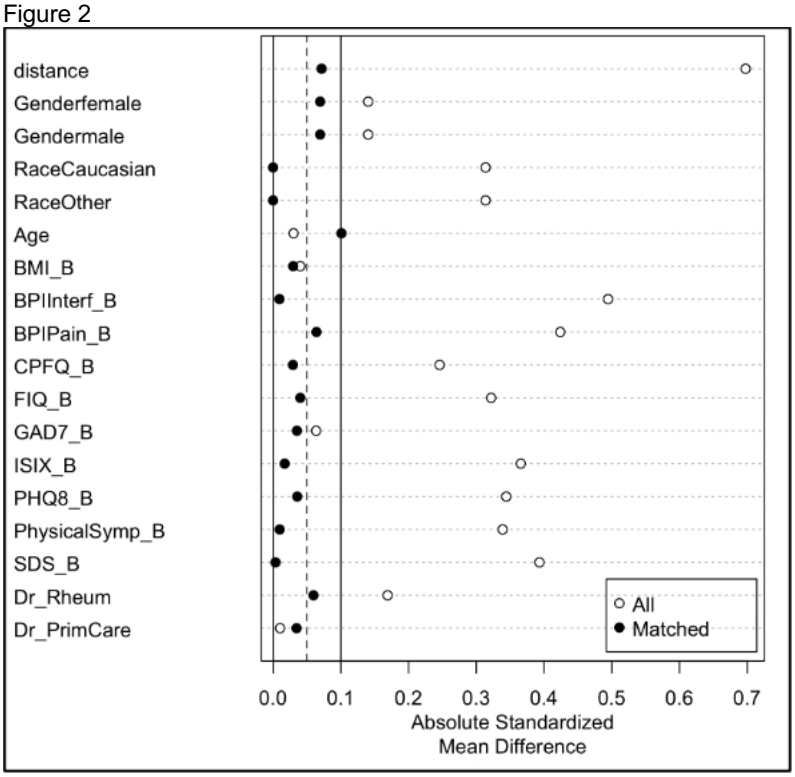


Figure 2: R Absolute Standardized Mean Difference Plot from plot()

PYTHON PSMPY

While Python (v: 3.8.5) is not widely used among real-world data researchers, its flexibility and ease of use is helping this language make its way into health analytics and should be considered a valid tool for PSM. There are limited options for propensity score matching modules in Python. Some have been created in the past and have become defunct, but luckily PsmPy (v: 0.3.13) was released in January 2023. The features and options are limited, which makes it a good choice for you to learn PSM modeling without getting overwhelmed with complex methods. This module only provides a K-Nearest Neighbor Matching algorithm so the results cannot be exactly compared to those from SAS and R, but coding and processes can be. Like R, Python requires extra coding to gather additional information and results from the PSM analysis. Here, the gender and dr_primcare distribution between the treatment groups got wider after matching – this algorithm would gain from exactly matching on each of these covariates. Below is the code used for this paper:

```
psm = PsmPy(refl3, treatment='cohort', indx='SubjID', exclude
        =['BPIPain_LOCF'])
   psm.logistic_ps(balance = True)
   psm.knn_matched(matcher='propensity_logit', replacement=False,
         caliper=None, drop_unmatched=False)
   psm.effect_size_plot(title='Standardized Mean differences across
         covariates before and after matching', save=False)
   matched_data = psm.matched_ids
```

Figure 3

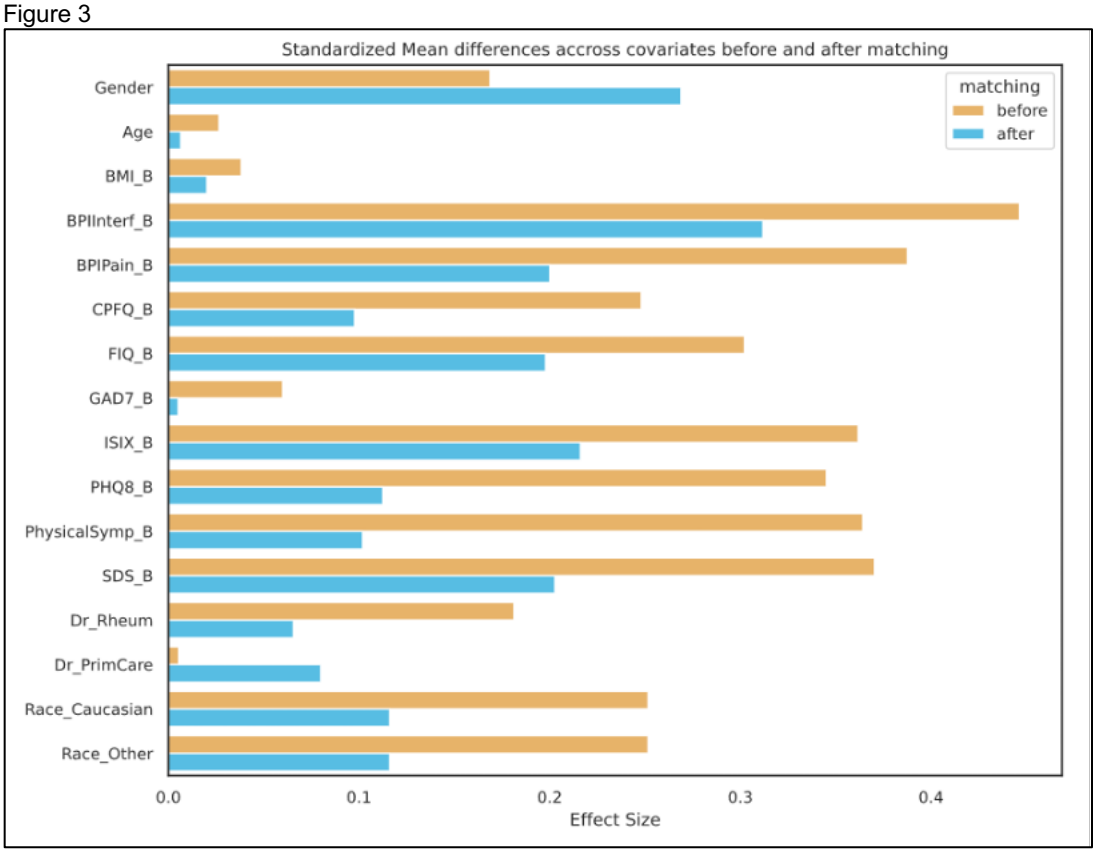


Figure 3: Python Absolute Standardized Mean Differences Plot from effect_size_plot()

INTERPRETING RESULTS AND CASUAL INFERENCE

At this stage, you can test the assumption that the outcome is the same between the two treatment arms with the test using only the matched subjects. The simplest statistical test is student's t-test, but additional methods can also be useful, including weighted linear regression and Complex Bootstrapping. Interpreting results of the former is the same as any t-test comparing means of two groups. Generally, we want to ensure that all assumptions are met first. Then you can determine if there's evidence for a difference in outcomes between the treatment groups based on tests for significance. For each of the matched datasets from the three languages, we have applied the following SAS code to model casual effect treatment, with the results displayed in each sub-heading, so only the matching algorithms are different among the three tests for causal effect. Below is the code used to determine casual effect:

```
proc ttest data=matched_data;
    class cohort;
    var BPIPain_LOCF;
    run;
```

SAS RESULTS

Below are the results from the PROC PSMATCH matched pairs. We see no significance difference between the two treatment options and generally normally distributed pain scores for each of the treatment arms from the Q-Q plot.

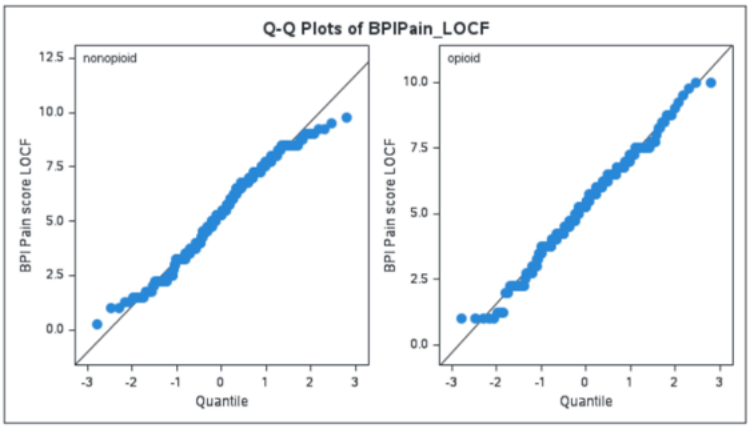
Output 1

The TTEST Procedure

Variable: BPIPain_LOCF (BPI Pain score LOCF)

cohort	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
nonopioid		238	5.3540	2.1333	0.1383	0.2500	9.7500
opioid		238	5.3235	1.8660	0.1210	1.0000	10.0000
Diff (1-2)	Pooled		0.0305	2.0041	0.1837		
Diff (1-2)	Satterthwaite		0.0305		0.1837		

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	474	0.17	0.8684
Satterthwaite	Unequal	465.75	0.17	0.8684



Output 1: SAS Casual Effect Results using PROC TTEST

R RESULTS

Below are the results from the MatchIt matched pairs. The algorithm was able to match all 240 subjects with opioid treatment to an appropriate subject. The p-values show that there is not a statistical different between the two treatment types on end of study pain measures and both groups have generally normal distributions.

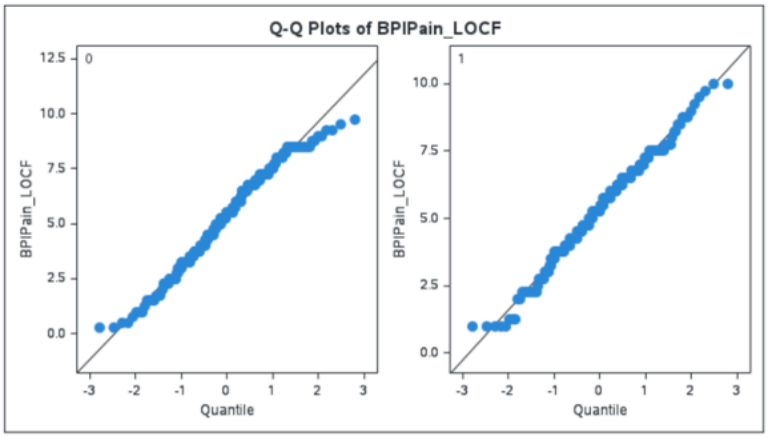
Output 2

The TTEST Procedure

Variable: BPIPain_LOCF

cohort	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
0		240	5.3229	2.1726	0.1402	0.2500	9.7500
1		240	5.3313	1.8651	0.1204	1.0000	10.0000
Diff (1-2)	Pooled		-0.00833	2.0247	0.1848		
Diff (1-2)	Satterthwaite		-0.00833		0.1848		

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	478	-0.05	0.9641
Satterthwaite	Unequal	467.29	-0.05	0.9641



Output 2: R Causal Effect Results using PROC TTEST

PYTHON RESULTS

Below are the Results from the PsmPy matched pairs. Here we see no evidence of a difference between final pain scores between the opioid and nonopioid treatment groups. The Q-Q plots also show relatively normally distributed outcome values.

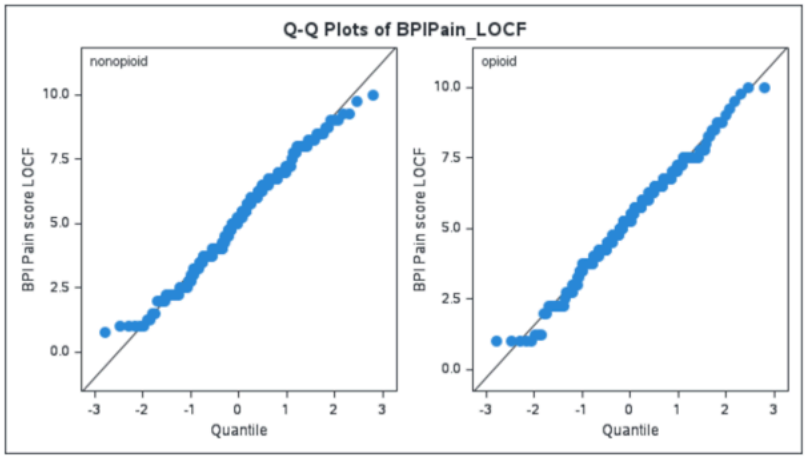
Output 3

The TTEST Procedure

Variable: BPIPain_LOCF (BPI Pain score LOCF)

cohort	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
nonopioid		238	5.1618	2.0485	0.1328	0.7500	10.0000
opioid		238	5.3162	1.8648	0.1209	1.0000	10.0000
Diff (1-2)	Pooled		-0.1544	1.9588	0.1796		
Diff (1-2)	Satterthwaite		-0.1544		0.1796		

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	474	-0.86	0.3903
Satterthwaite	Unequal	469.88	-0.86	0.3903

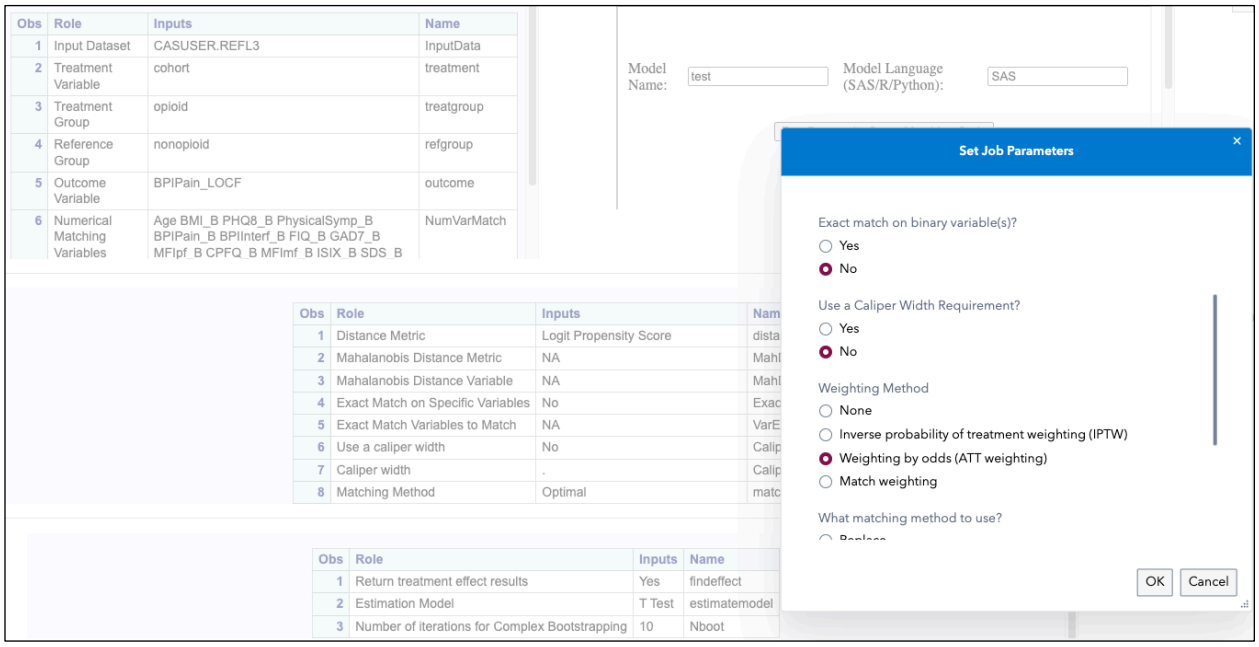


Output 3: Python Causal Effect Results from PROC TTEST

RESULTS IN SAS VIYA VISUAL ANALYTICS

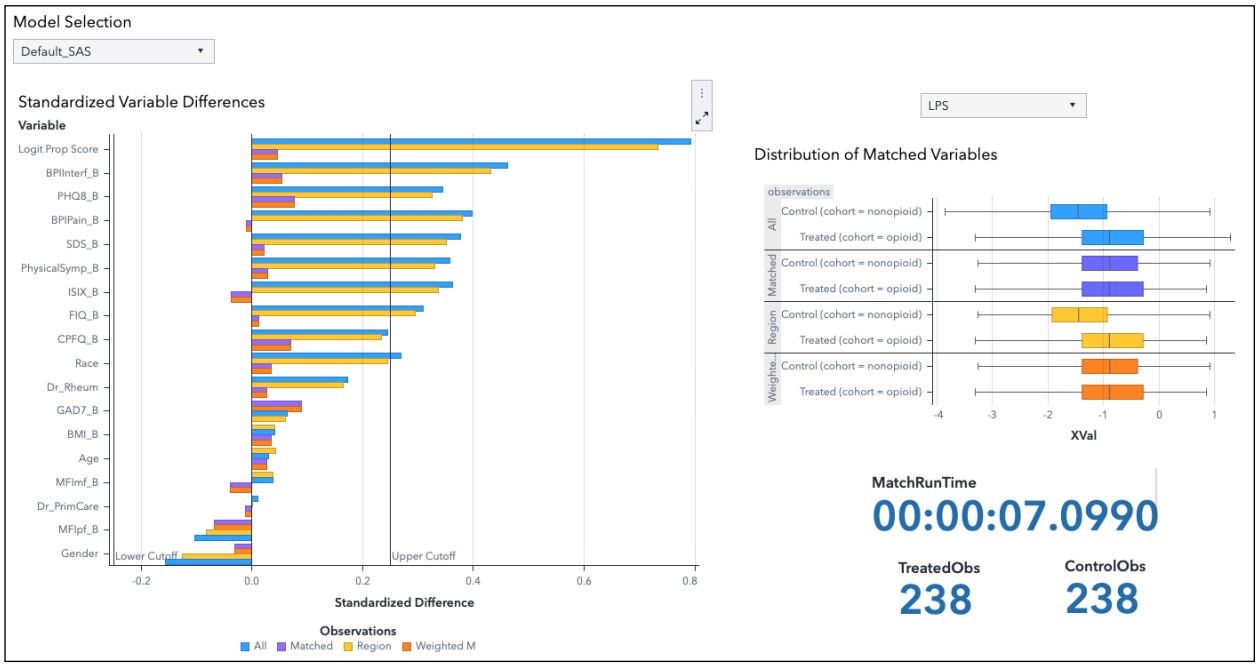
All the analyses in this paper were run in SAS Studio on Viya. There are many details and options for running PSM analyses and keeping track of all the different results can be cumbersome. Comparing the default plots and outputs from each of the languages adds extra time to determining the best model for the study. Below are screenshots of an interactive SAS Visual Analytics dashboard that eliminates the need to switch between platforms and outputs. The first step is to select all the matching parameters and run the process in either SAS, Python, or R (keeping in mind PsmPy does not have all features available). You can change these to whatever specifications without any coding required. Each named model populates the second dashboard to show distributions of variables and standardized mean differences before and after matching. We chose to display the run time of the matching algorithm and the number of treated and control observations in the matched group. Selecting on the model will change the plots and values so you can immediately see how different matching algorithms and features change the results. This allows anyone without coding knowledge to access and choose the best matching algorithm for their study quickly and accurately.

Display 1



Display 1: Screenshot of selecting job parameters for PSM analysis

Display 2



Display 2. Screenshot of interactive display of PSM results

CONCLUSION

Propensity score matching gives you the opportunity to overcome challenges of selection bias in observational studies by balancing the likelihood of receiving a specific treatment. What RCTs can address implicitly, PSM can attend to statistically. You can construct comparable study groups to gain a better understanding of casual relationships between intervention and outcome. We can reach into evidence-based care from data previously unavailable for such purpose. Python's module PsmPy is a

good choice if all the options and nuances in the other languages overcomplicates a cursory check to see if PSM can work for their study objectives. The MatchIt library in R has the option to use one of multiple ML algorithms to calculate the propensity score without extra coding. This is a good option for those looking to test the effect of specific ML calculations of scores on matching. SAS's procedure, PROC PSMATCH, provides comprehensive numerical and graphical outputs by default. PROC PSMATCH is useful for researchers interested in creating reports efficiently. Given the consistency in conclusions across the three algorithms, the final goal of improving patient outcomes through PSM analysis is not limited to any specific language. Rather, we recommend using the tool that best allows for reliable and reproducible results for each study and researcher individually. SAS VIYA and Visual Analytics provides a platform to run and compare all results quickly and reliably to best inform next steps. Robust, reproducible, and reliable results are required for real-world data analytics and SAS Viya provides the tools necessary for these studies. Propensity score matching can advance our understanding of causal relationships using RWD and all work done to provide better care for patients is a step forward. Using PSM in health data analysis fortifies the robustness of all research and the flexibility of multi-lingual algorithms ensures accessibility and applicability across many studies and institutions.

REFERENCES

Benedetto U., Head S.J., Angelini G.A., & Blackstone E.H. (2018). Statistical primer: propensity score matching and its alternatives, *European Journal of Cardio-Thoracic Surgery*. 53(6):1112–1117.

Daniel Ho, Kosuke Imai, Gary King, & Elizabeth Stuart. (2007). "Matching as Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference." *Political Analysis*, 15: 199–236.

Faries D., Zhang X., Kadziola Z., Siebert U., Kuehne F., Obenchain R. L., & Haro J. M. (2020). *Real-World Health Care Data Analysis*. SAS Institute.

Franklin, J. M., Schneeweiss, S., Polinski, J. M., & Rassen, J. A. (2014). Plasmode simulation for the evaluation of pharmacoepidemiologic methods in complex healthcare databases. *Computational statistics & data analysis*, *72*, 219–226.

Gadbury GL, Xiang Q, Yang L, Barnes S, Page GP, & Allison DB (2008) Evaluating Statistical Methods Using Plasmode Data Sets in the Age of Massive Public Databases: An Illustration Using False Discovery Rates. PLoS Genet 4(6): e1000098.

Groenwold R. H. H. (2020). Commentary: Quantifying the unknown unknowns. *International journal of epidemiology*, *49*(5), 1503–1505.

Ho D, Imai K, King G, & Stuart E (2011). "MatchIt: Nonparametric Preprocessing for Parametric Causal Inference." *Journal of Statistical Software*, 42(8), 1-28. doi:10.18637/jss.v042.i08

Kline, A., & Luo, Y. (2022). PsmPy: A Package for Retrospective Cohort Matching in Python. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, *2022*, 1354–1357.

Robinson, R. L., Kroenke, K., Mease, P., Williams, D. A., Chen, Y., D'Souza, D., Wohlreich, M., & McCarberg, B. (2012). Burden of illness and treatment patterns for patients with fibromyalgia. *Pain medicine (Malden, Mass.)*, *13*(10), 1366–1376.

Rosenbaum, P. R., & Rubin, D. B. (1983). The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Biometrika*, *70*(1), 41–55.

Rosler A., Constantin G., Nectoux P., Ferreira G., Borges A., Sales M.C., & Lucchese, F.A. (2020). Impact of atrial fibrillation on in-hospital outcomes of coronary artery bypass graft surgery: an analysis by propensity score matching, *European Heart Journal*, Volume 43, Issue Supplement_2, ehac544.2162.

Zhao, Q. Y., Luo, J. C., Su, Y., Zhang, Y. J., Tu, G. W., & Luo, Z. (2021). Propensity score matching with R: conventional methods and new features. *Annals of translational medicine*, *9*(9), 812.

ACKNOWLEDGMENTS

Thank you to all my colleagues at SAS.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Catherine Briggs
SAS Institute, Inc.
cat.briggs@sas.com