

Comparing the Performance of Propensity Score Matching Algorithms from Multiple Programming Languages

By Catherine Briggs & Sherrine Eid
February 26, 2024



Agenda

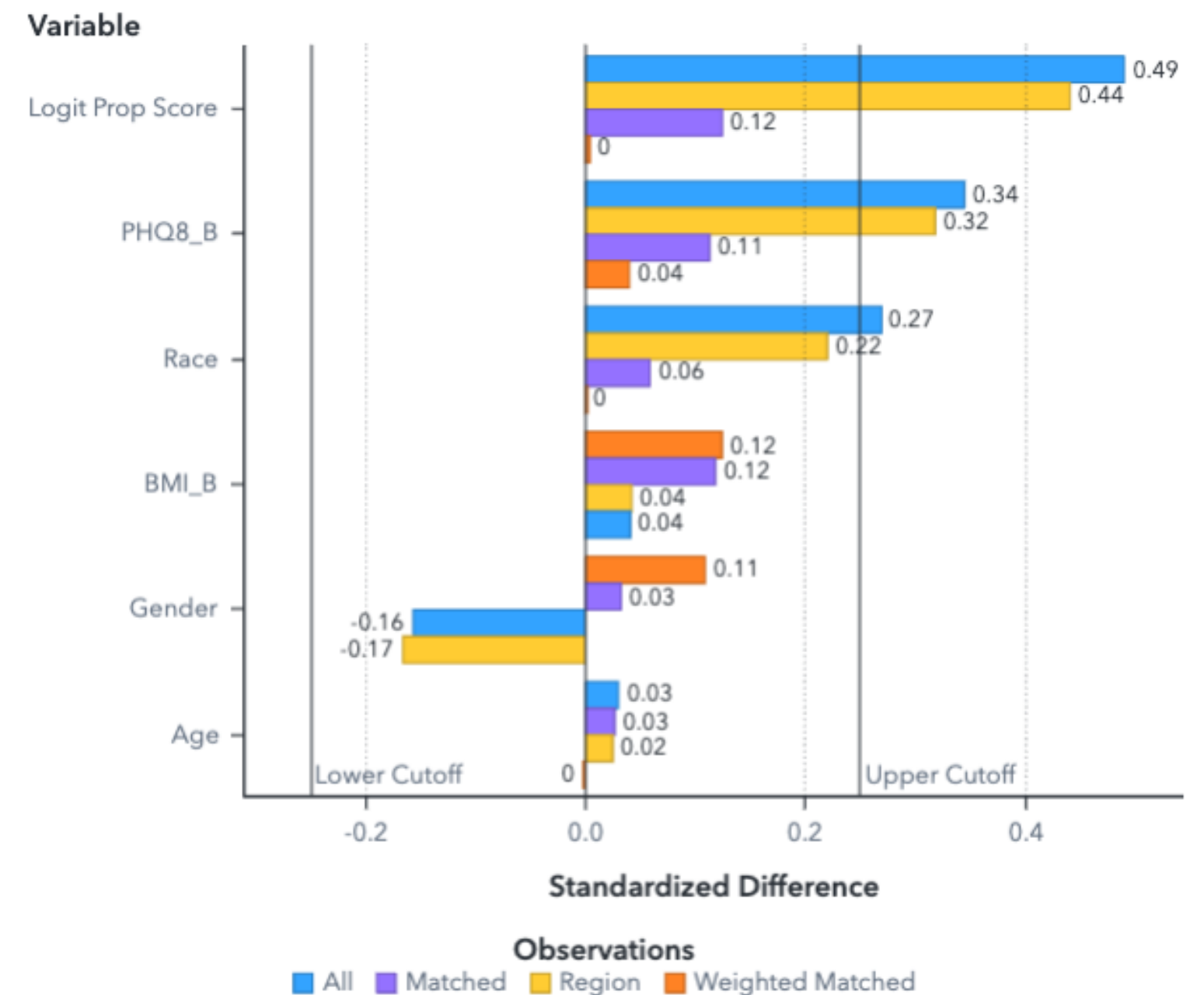
- Propensity Score Matching In Real World Evidence
- Coding for PSM
- Interpreting Results
- Using PSM in Casual Inference
- Conclusion
- Questions

Propensity Score Matching In Real World Evidence

Propensity Score Matching Overview

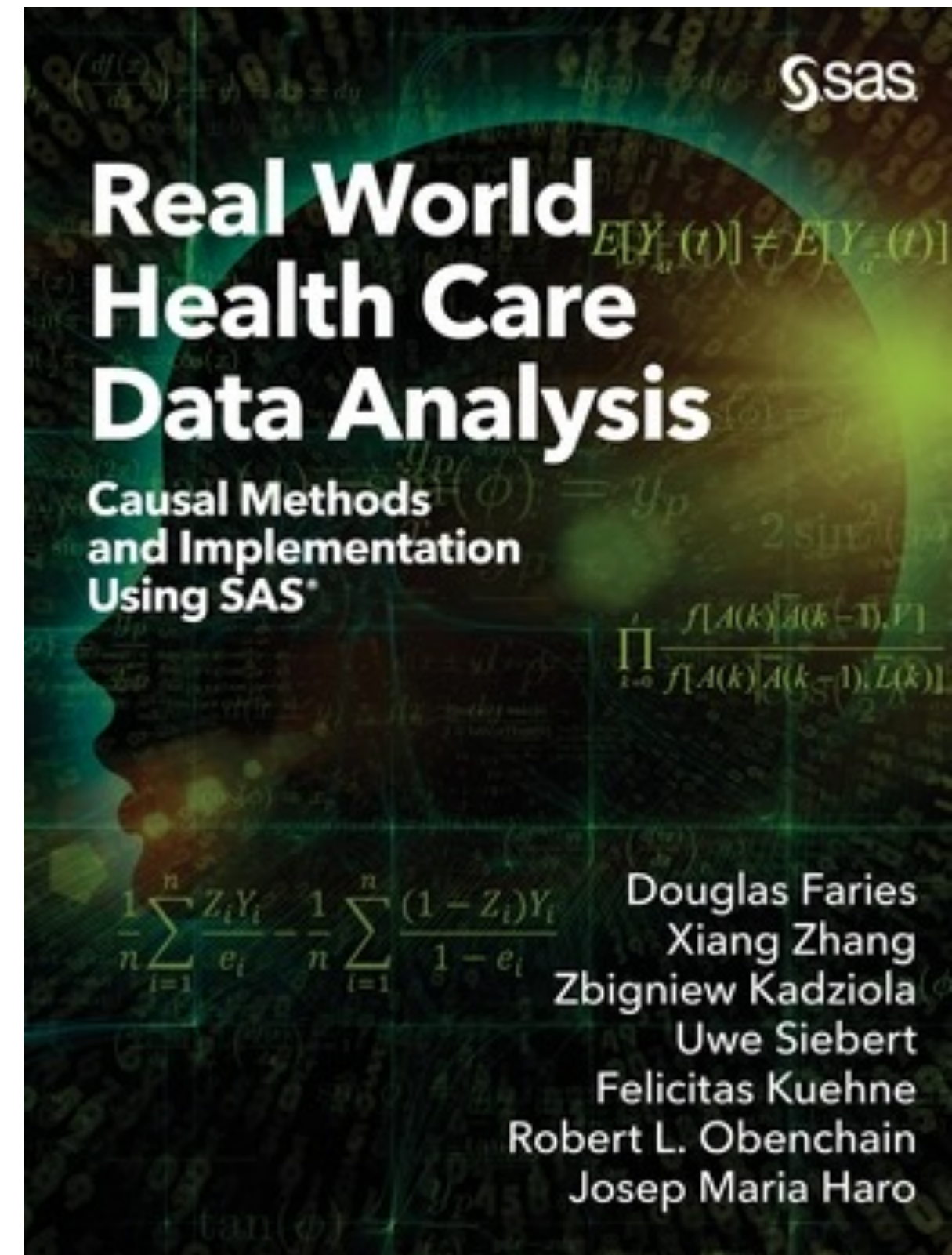
- Statistical method to mimic randomization when it isn't possible
- Used to analyze observational data to obtain unbiased causal treatment effect estimates
- The Propensity Score is a collection of information for covariates into one number for each subject

Standardized Variable Differences



PS Match Use Case

- Data for this use case is a simulated dataset based on an observational fibromyalgia study
 - Goal is to estimate the casual treatment effect of opioid vs. nonopioid medication on final pain scores
- The book, and it's outlined processes, is an excellent resource for PSM and causal inference



Coding for Propensity Score Matching

SAS PROC PSMATCH; R MatchIt; Python PsmPY

Universal Key Components

- Propensity Score Model
 - Statistical model using pre-treatment covariates to find the likelihood of each subject to receive their recorded treatment
 - Standard is a logistic regression, but can be more advanced ML (i.e., decision tree, neural network)
- Perform Matching
 - Multiple algorithms exist to match observations between the control and treatment groups
 - Minimize distance between two observations
 - ‘Optimal’ is the default method for both SAS PROC PSMATCH and R MatchIt
- Assess Balance
 - Determine if treatment and control groups are adequately matched
 - Re-match with exact variables or using another method

SAS: PROC PSMATCH

Code Breakdown

```
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(obs=MATCH)=matched_data;  
  
run;
```

Propensity Score Model

Find Matches

Assess Balance

Save Matches

R: MatchIT

Code Breakdown

Get PS Score and Match Observations

```
matched <- matchit(cohort ~  
Gender+Race+Age+BMI_B+BPIInterf_B+BPIPain_B+CPFEQ_B+FIQ_B+G  
AD7_B+ISIX_B+PHQ8_B+PhysicalSymp_B+SDS_B+Dr_Rheum+Dr_Pri  
mCare,
```

```
data = refl3,  
distance = "glm",  
link = "logit",  
method = "optimal",  
ratio = 1,  
caliper = NULL)
```

Find matches

Propensity Score Model

Plot Standardized Mean Differences

```
plot(summary(matched, un=TRUE))
```

Assess Balance

PSM Results -- Data Set with all covariates, matches, and matching metric

```
results <- summary(matched, un=TRUE)
```

```
matched_data <- match.data(matched, include.s.weights =  
TRUE, drop.unmatched = FALSE)
```

Save Matches

Python: PsmPy

Code Breakdown

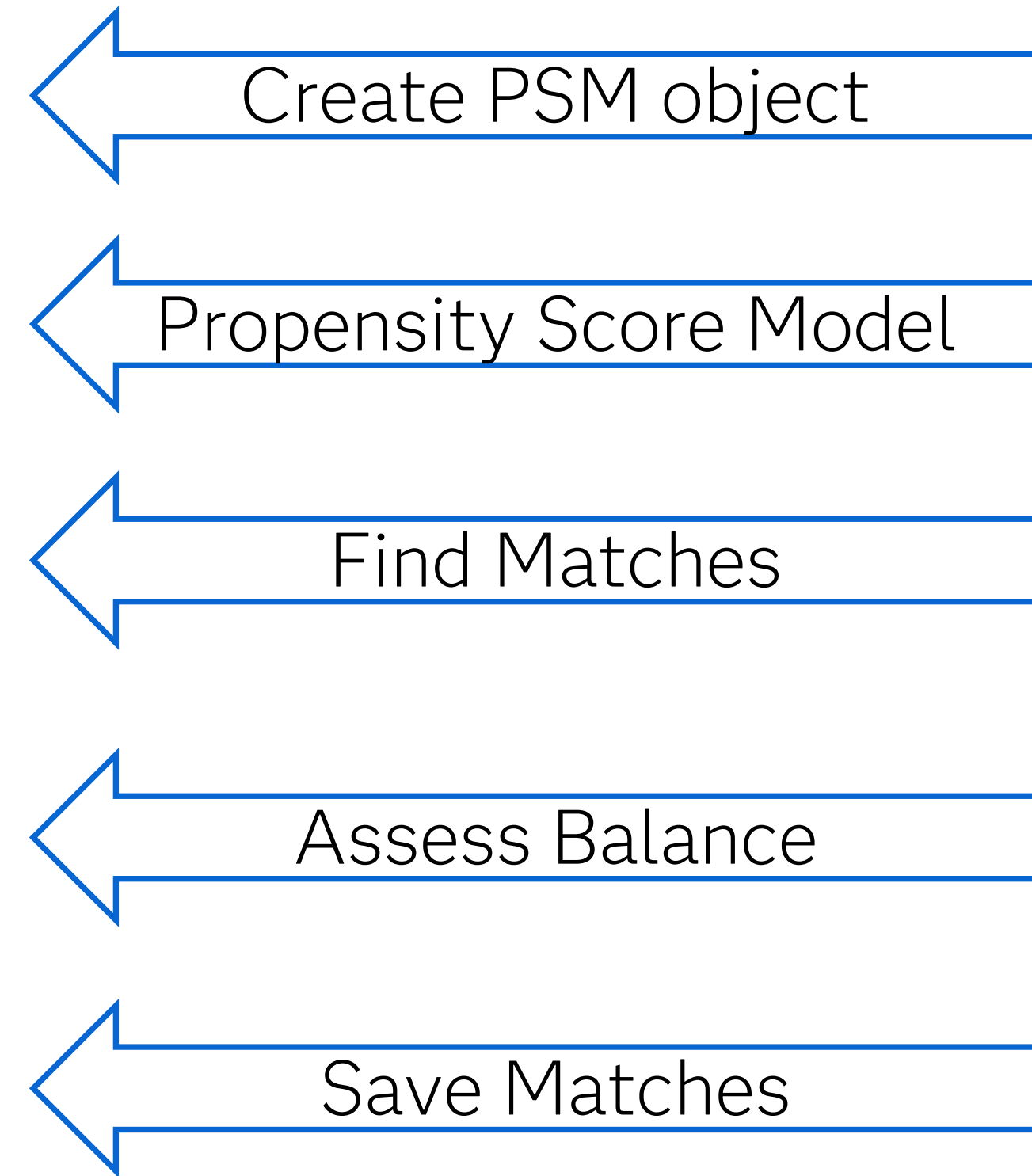
```
psm = PsmPy(refl3, treatment='cohort',  
            indx='SubjID', exclude = ['BPiPain_LOCF'])
```

```
psm.logistic_ps(balance = True)
```

```
psm.knn_matched(matcher='propensity_lo  
git', 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
```



Create PSM object

Propensity Score Model

Find Matches

Assess Balance

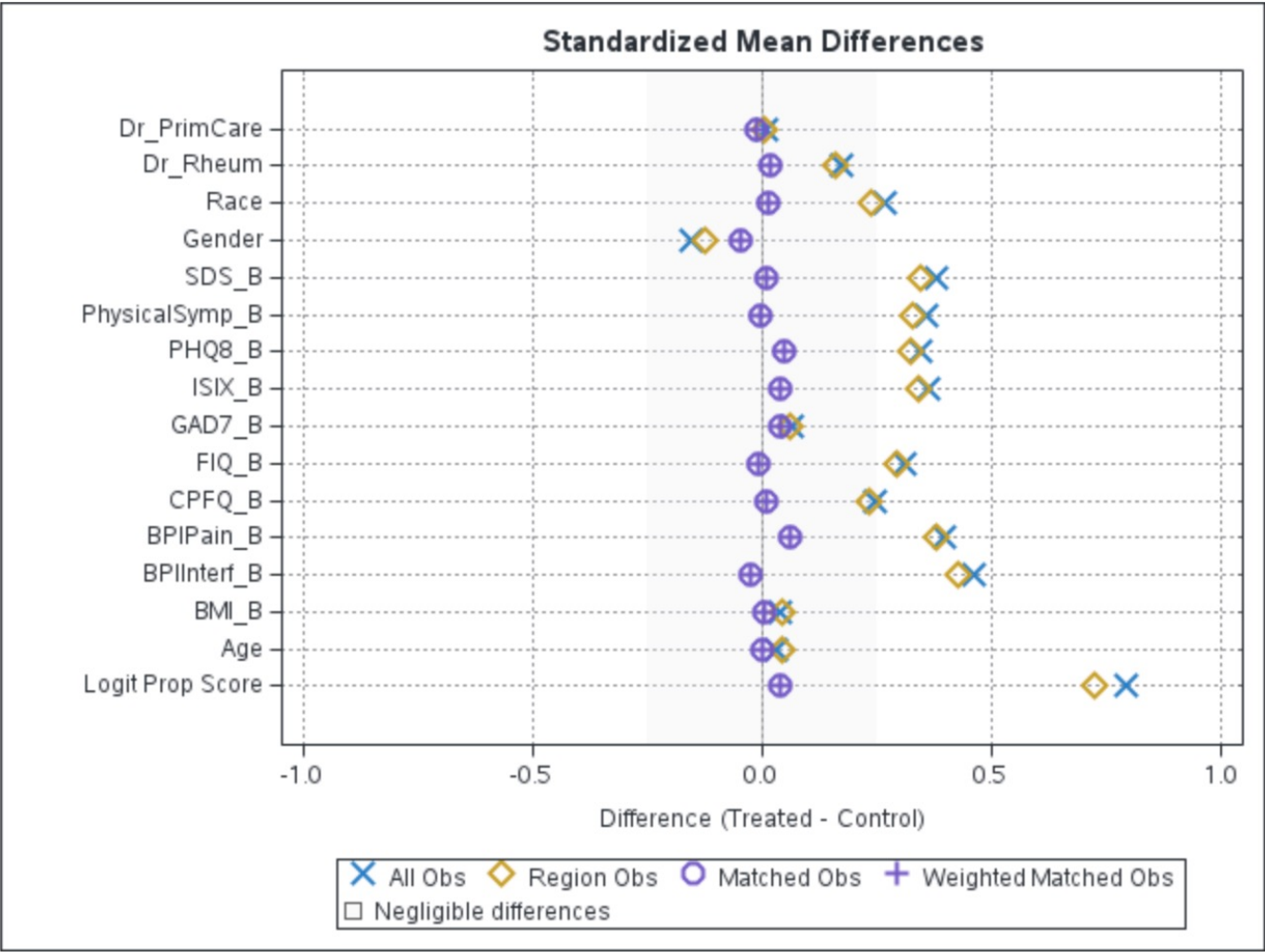
Save Matches

Best Practices

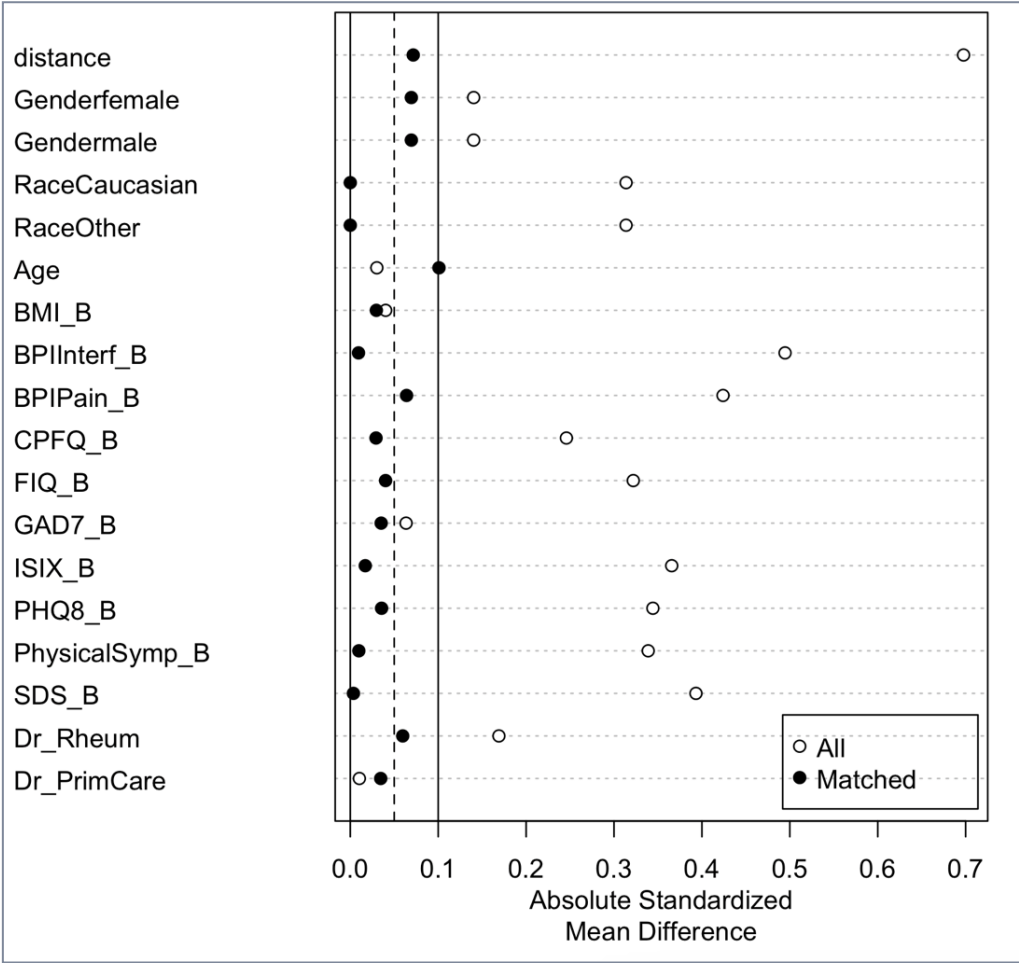
- Remove observations with missing outcome before matching
 - Must be dropped from causal inference analysis, which unbalances the groups
- Start with simplest matching options before testing more complex features
 - Initial balanced matched to compare new methods
 - Errors are easier to understand and fix
- Consider the trade-off between perfectly balanced groups and cohort size

Interpreting Results

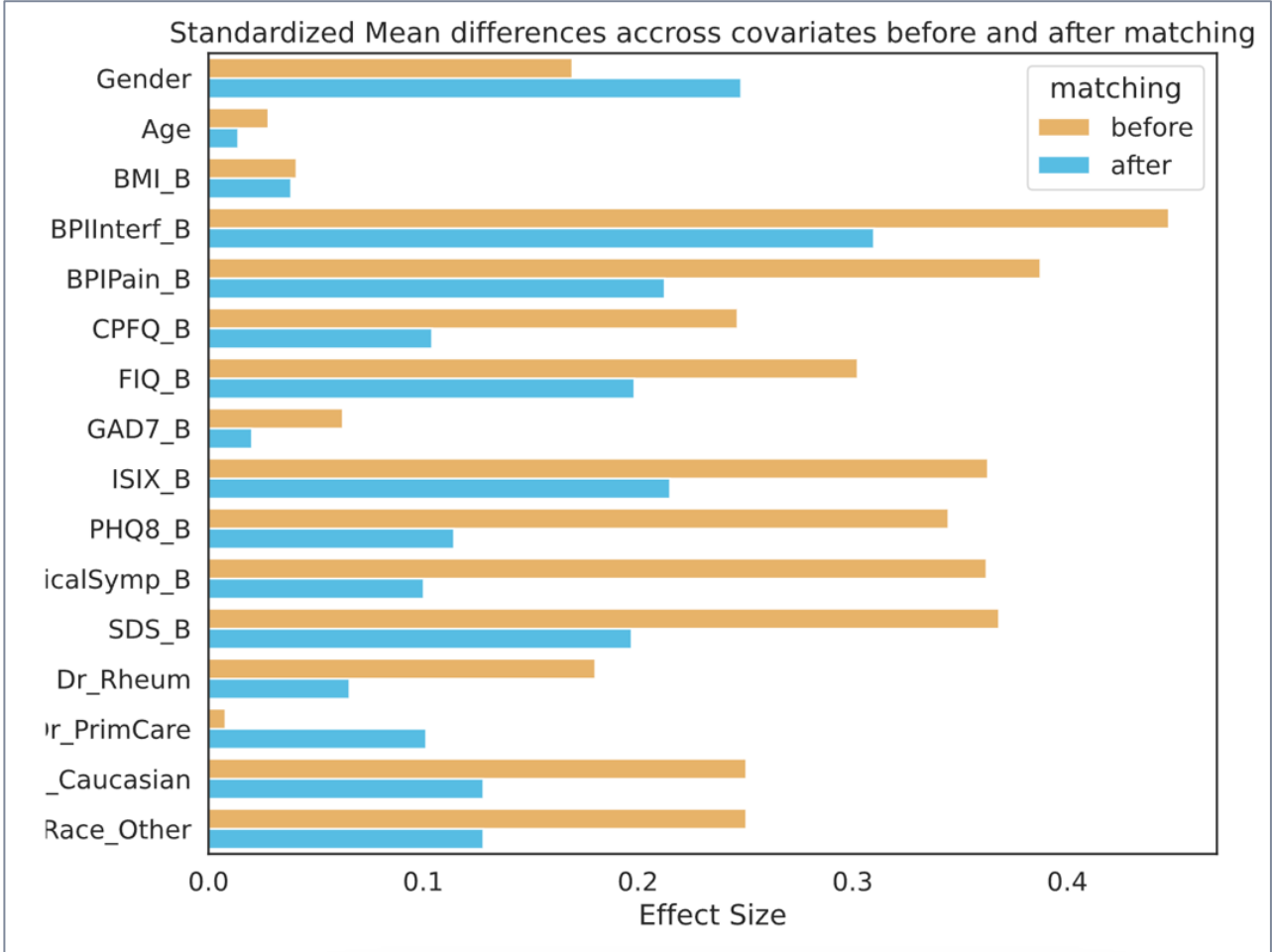
Results from Default Options



Proc PSMATCH



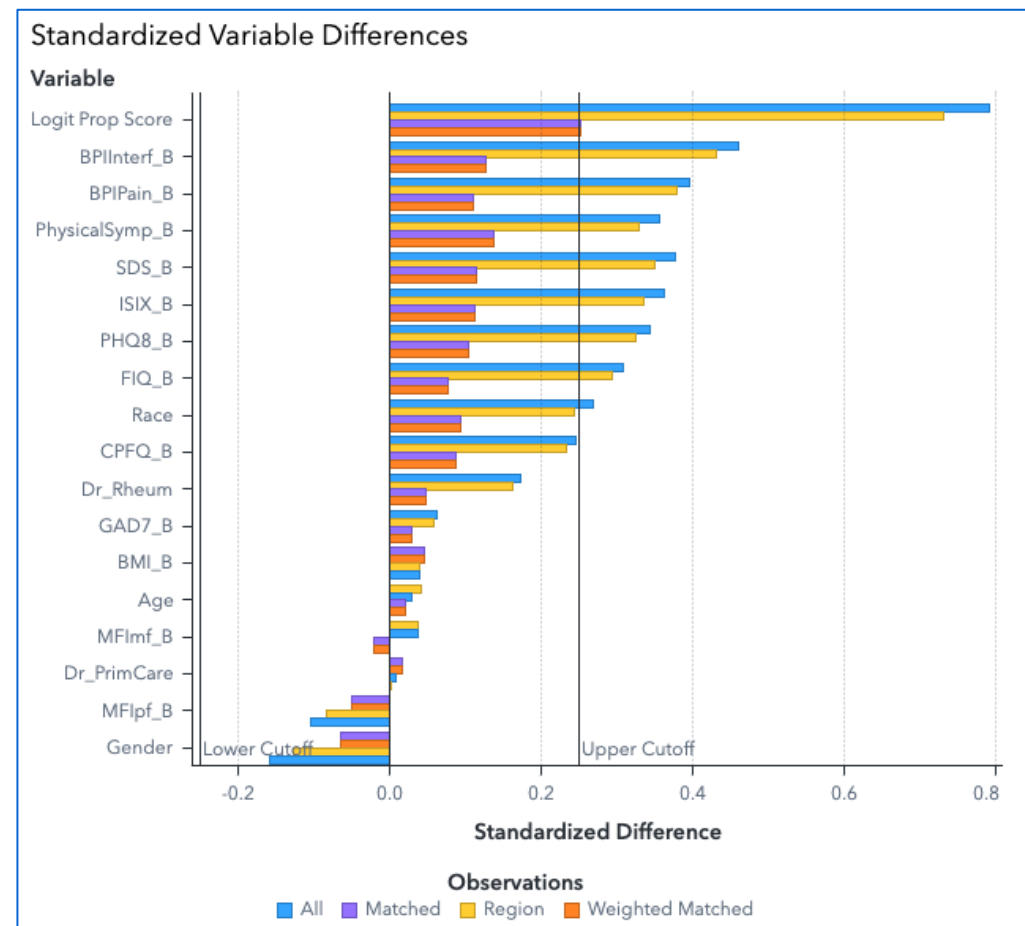
R MatchIt



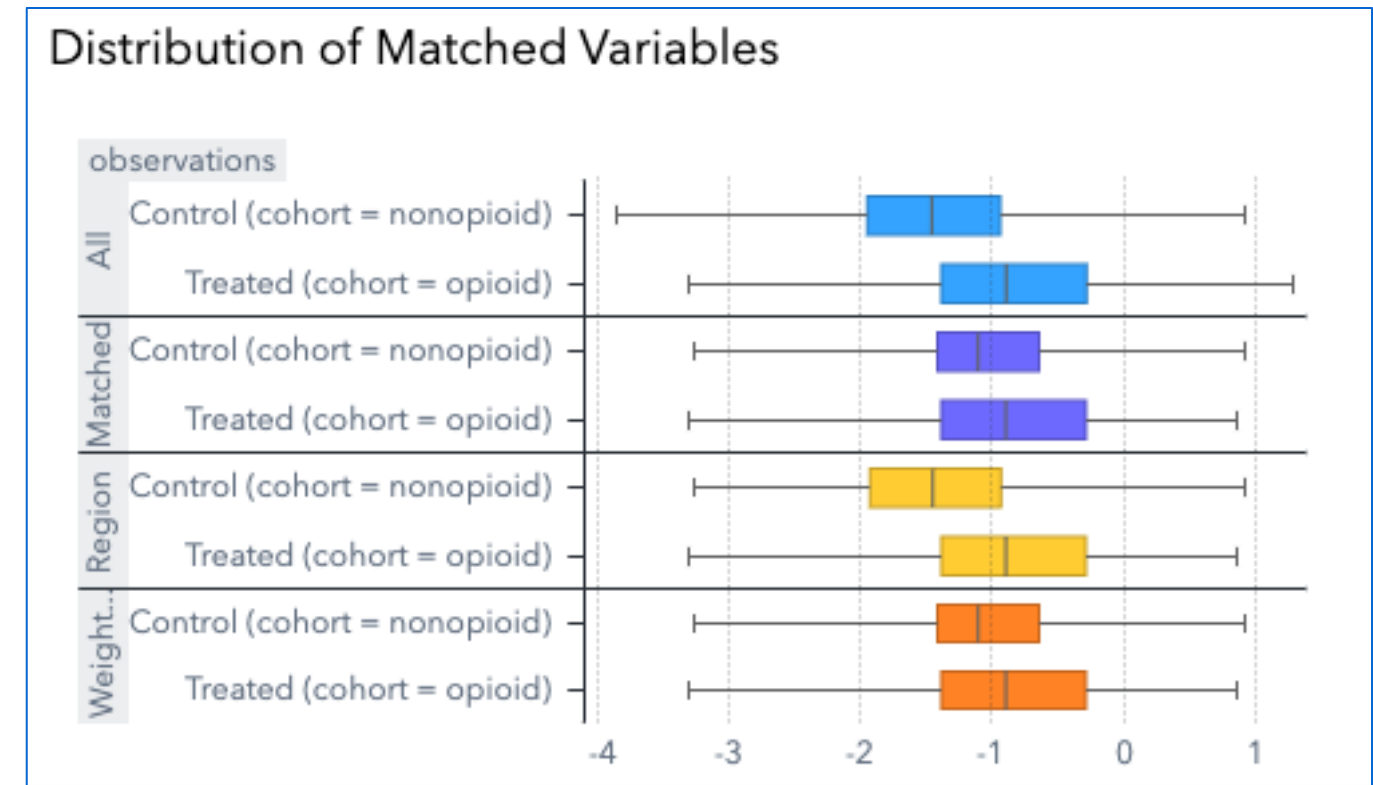
Python PsmPy

What to look for

Assessing balance and Observation Size



Determine if variable differences are within threshold



Look for equal distributions for each covariate

ControlObs	TreatedObs
476	238

Minimize observation loss

Using PSM for Casual Inference

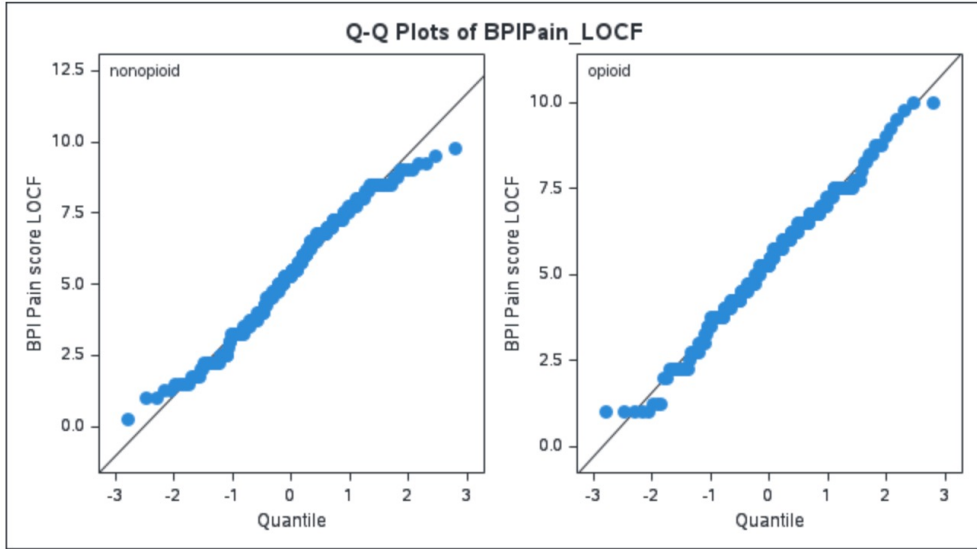
Universal Key Components

- Determine best method for estimating effect of treatment on response
 - T-test is simplest; can use other methods including weighted regression and bootstrapping
- Ensure statistical assumptions are met for chosen method
 - Normal distribution, continuous data, etc. for T-test
- Examine p-value for hypothesis test

Results from Default Options

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		

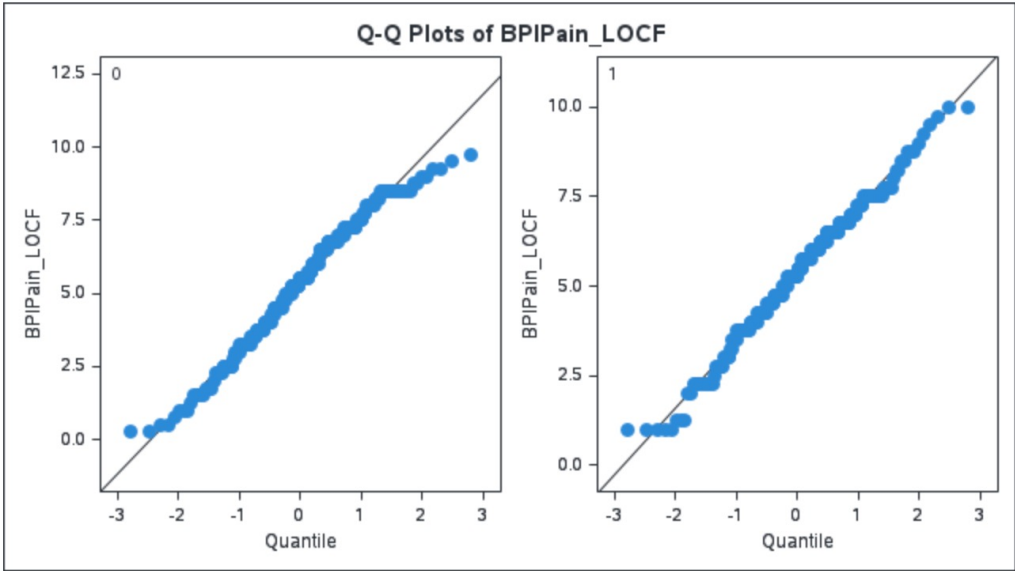
Method	Variances	DF	t Value	Pr > t
Pooled	Equal	474	0.17	0.8684
Satterthwaite	Unequal	465.75	0.17	0.8684



Proc PSMATCH

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		

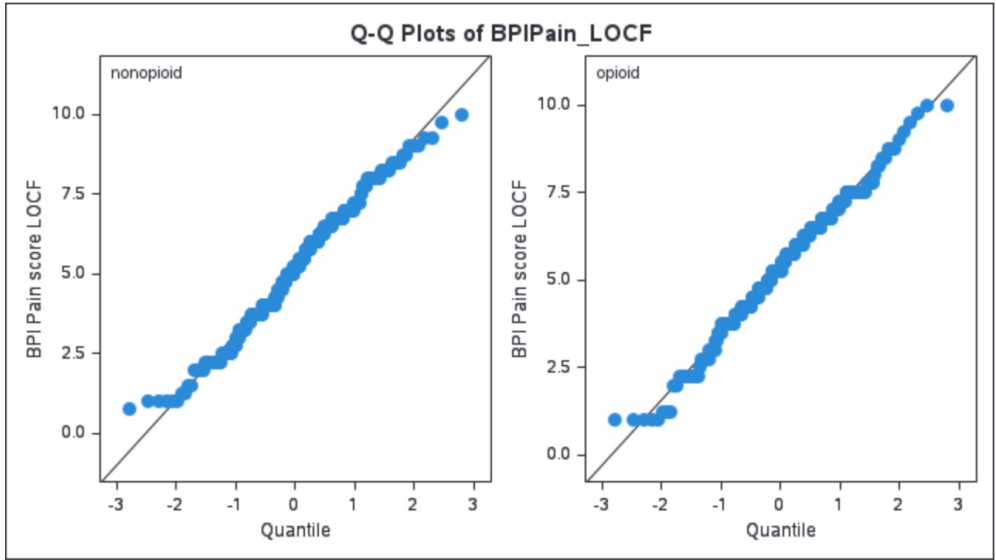
Method	Variances	DF	t Value	Pr > t
Pooled	Equal	478	-0.05	0.9641
Satterthwaite	Unequal	467.29	-0.05	0.9641



R MatchIt

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



Python PsmPy



Conclusion

Conclusion

Which Language is Best?

- SAS PROC PSMATCH
 - Automatic visual outputs for balance assessment
- R MatchIt
 - Option for non-regression ML propensity score modeling
- Python PsmPy
 - Simple implementation for new researchers

PSM in RWD

- Remove selection bias from observational studies
- Can influence research-based care and inform new areas of study
- Use in place in of an unethical RCT

Questions?

cat.briggs@sas.com; sherrine.eid@sas.com

