

Oregon Department of Environmental Quality

An R-based Web Application to Search, Analyze and Display Water Quality Data in Oregon State, USA

Peter Bryant

May 4th, 2016
National Water Quality Monitoring Council
10th National Monitoring Conference
Tampa, Florida

Web Application Goals

Provide an interactive means for users to:

- Query data from multiple databases
- Evaluate status and trend at individual sampling stations
- Display the results

Non-R users are able to generate information on status and trends

Water Quality Data Start Here...



...And Go Here

- DEQ databases – Current and legacy databases for grab and continuous data



- NWQMC Water Quality Portal for grab data



– EPA STORET



– USGS NWIS



- USGS National Water Information System for continuous data



Then What?

Use the data to:

- Determine status of water quality
- Look for trends in water quality

But How?

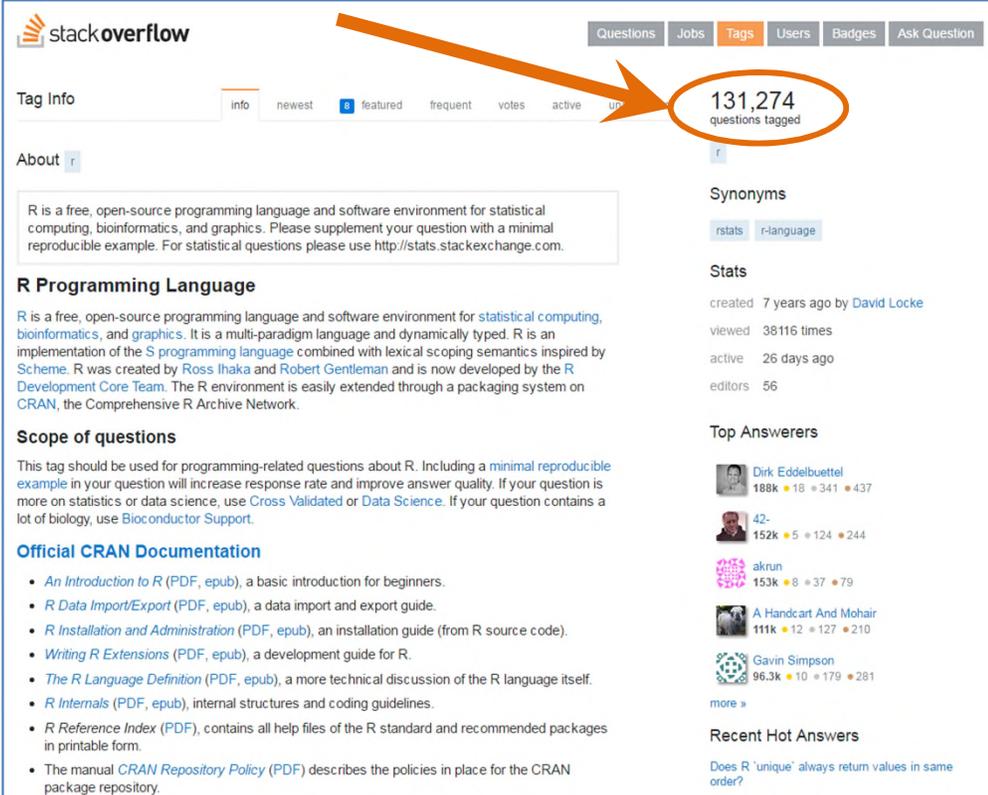
That's a lot of places to look and potentially a lot of data to analyze

Luckily there's



What is ?

- Open source statistical programming language
- It's free (r-project.org)
- Has a large user base
- Lots of documentation and learning resources



The screenshot shows the Stack Overflow interface for the 'r' tag. The top navigation bar includes 'Questions', 'Jobs', 'Tags', 'Users', 'Badges', and 'Ask Question'. The 'Tags' tab is active. Below the navigation bar, the 'Tag Info' section shows '131,274 questions tagged' circled in orange, with an orange arrow pointing to it from the top right. The 'About r' section provides a brief description of R as a free, open-source programming language and software environment for statistical computing, bioinformatics, and graphics. It mentions that R is a multi-paradigm language and dynamically typed, and is an implementation of the S programming language. The 'R Programming Language' section includes a detailed description of R and its development. The 'Scope of questions' section provides guidelines for using the tag. The 'Official CRAN Documentation' section lists several resources, including 'An Introduction to R', 'R Data Import/Export', 'R Installation and Administration', 'Writing R Extensions', 'The R Language Definition', 'R Internals', 'R Reference Index', and 'The manual CRAN Repository Policy'. The 'Stats' section shows that the tag was created 7 years ago by David Locke, viewed 38116 times, and is active for 26 days. The 'Top Answerers' section lists several users, including Dirk Eddelbuettel, 42-, akrun, A Handcart And Mohair, and Gavin Simpson. The 'Recent Hot Answers' section shows a question about the 'unique' function.

How Does Work?

- Use a text editor to write out the analysis
- Pass the analysis code to the R software to actually run the analysis
- R users have built packages to do many different types of analysis
- A package is a collection of functions to perform an analysis

The Shiny Package

- The Shiny package is built by  R Studio
 - Provides R functions that translate R code to an interactive javascript web application
 - Builds a user interface to edit settings in the analysis
 - Puts all the code to run an analysis behind the user interface

While I see this

```
96
97
98 if ('Water Quality Portal' %in% input$db) {
99   incProgress(1/10, detail = 'Querying the water quality Portal')
100   prog <- prog + 1/10
101   wqpData <- tryCatch(wqpQuery(planArea = input$select,
102                               HUCList = HUCList,
103                               inParms = input$parms,
104                               luParms = parms,
105                               startDate = input$dates[1],
106                               endDate = input$dates[2]),
107                       error = function(err) {err <- geterrmessage()})
108
109   if (any(c('Temperature', 'pH') %in% input$parms)) {
110     incProgress(1/10, detail = 'Querying NWIS continuous data')
111     prog <- prog + 1/10
112     nwisData <- tryCatch(nwisQuery(planArea = input$select,
113                                   HUCList = HUCList,
114                                   inParms = input$parms,
115                                   startDate = input$dates[1],
116                                   endDate = input$dates[2]),
117                           error = function(err) {err <- geterrmessage()})
118   }
119
120   if (is.null(wqpData) & is.null(nwisData)) {
121     wqp_message <- 'Your query returned no results from the water quality Portal.'
122   } else if (!is.data.frame(wqpData) & !is.null(wqpData)) {
123     if (grepl("307", wqpData)) {
124       wqp_message <- 'water quality Portal is busy. Please try again in a few minutes.'
125     }
126   }
127
128   if ('DEQ' %in% input$db) {
129     incProgress(1/10, detail = 'Querying the LASAR database')
130     prog <- prog + 1/10
131
132     lasarData <- lasarQuery(planArea = input$select,
133                             HUCList = HUCList,
134                             inParms = input$parms,
135                             startDate = input$dates[1],
136                             endDate = input$dates[2])
137
138     odbcCloseAll()
139     if (nrow(lasarData) == 0) lasarData <- NULL
140
141     incProgress(1/10, detail = 'Querying the Element database')
142     prog <- prog + 1/10
143
144     elmData <- elementQuery(planArea = input$select,
145                              HUCList = HUCList,
146                              inParms = input$parms,
147                              startDate = input$dates[1],
148                              endDate = input$dates[2])
149
150     odbcCloseAll()
151     if (nrow(elmData) == 0) elmData <- NULL
152   }
153 }
```

```
17 #We want to extract only those stations in the current AgwQMA so let's bring that lay
18 agwqma <- readoGR(dsn = './data/GIS', layer = 'ODA_AgwQMA', verbose = FALSE)
19
20 shinyUI(fluidPage(
21
22   titlePanel("Oregon Water Quality Status and Trend Beta Version 2.0"),
23
24   mainPanel(
25     HTML("<script> if (!window.chrome) { alert('For full functionality you will need
26     tabsetPanel(
27       tabPanel("Data query", fluidRow(
28         column(3,
29           selectInput("select", label = h3("select Plan Area"),
30                       #choices = list()
31                       choices = c("Choose one" = "", sort(agwqma$PlanName)))
32         ),
33         column(3,
34           checkboxGroupInput("parms", label = h3("select Paramters to Query"),
35                               choices = c('Temperature', 'pH', 'Bacteria'),
36                               selected = 1)
37         ),
38         column(3,
39           dateRangeInput("dates", label = h3("select the start and End Dates")))
40       ),
41     ),
42     fluidRow(
43       column(3,
44         checkboxGroupInput('db', 'select Database(s) to Query:',
45                             c('Water Quality Portal', 'DEQ'),
46                             selected = 1)
47       ),
48       column(3,
49         h3("Run Query"),
50         actionButton(inputId = "action_button", label = 'submit')
51       ),
52     ),
53     fluidRow(
54       column(3,
55         column(3,
56           h3(" "),
57           htmlOutput("text1"),
58           h3(" "),
59           htmlOutput("text2")
60         )
61       ),
62     ),
63     fluidRow(
64       column(3,
65         tableOutput('all_totals'),
66       ),
67       column(3,
68         conditionalPanel(condition = "output.text2 == ''",
69                           "Click here to download the data",
70                           downloadButton('downloadData', 'download')
71                           uiOutput('downloadData')
72       )
73     ),
74   )
75 )
```

You see this

Oregon Water Quality Status and Trend Beta Version 2.0

Data Query | Review Data | Plot Status and Trend

Select Plan Area: Burnt River

Select Parameters to Query: Temperature, pH, Bacteria

Select the Start and End Dates: 2006-04-05 to 2016-04-05

Select Database(s) to Query: Water Quality Portal, DEQ

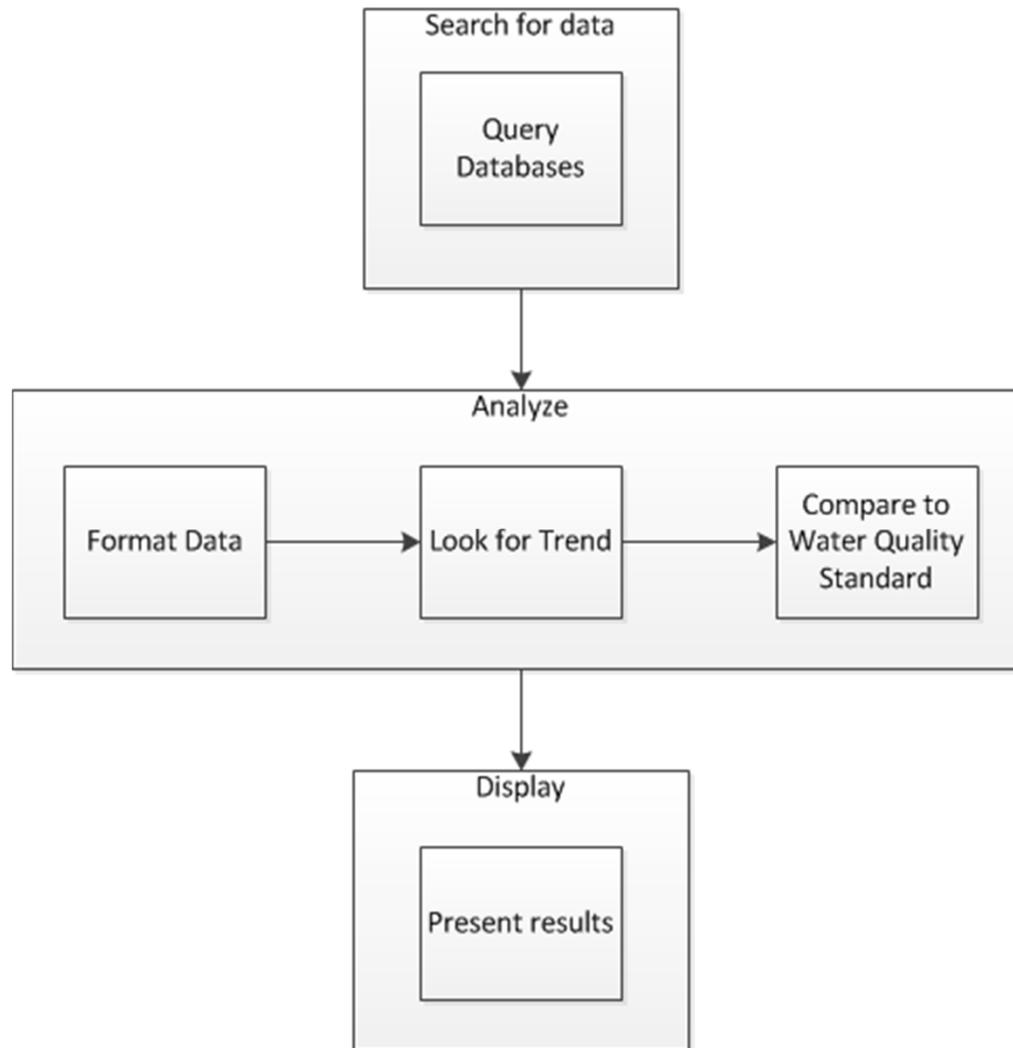
Run Query: Submit

You just submitted Burnt River Plan Area Query for Temperature, pH, Bacteria from 2006-04-05 to 2016-04-05

	Analyte	Stations	Results
1	E. Coli	16	502
2	Fecal Coliform	4	12
3	pH	23	7376
4	Temperature	43	94621

Download the data | View map

Application Structure



Search For Data

Query Databases

- RODBC package for DEQ data
 - Provides connection from R to SQL Server databases
- dataRetrieval package for Water Quality Portal data
 - A collection of functions to get data from USGS and EPA National Databases
 - Uses web services to communicate with the databases

Search For Data

Oregon Water Quality Status and Trend Beta Version 2.0

Data Query

[Review Data](#)

[Plot Status and Trend](#)

Select Plan Area

Burnt River

Select Parameters to Query

Temperature

pH

Bacteria

Select the Start and End Dates

2006-04-05

to

2016-04-05

Select Database(s) to Query:

Water Quality Portal

DEQ

Run Query

Submit

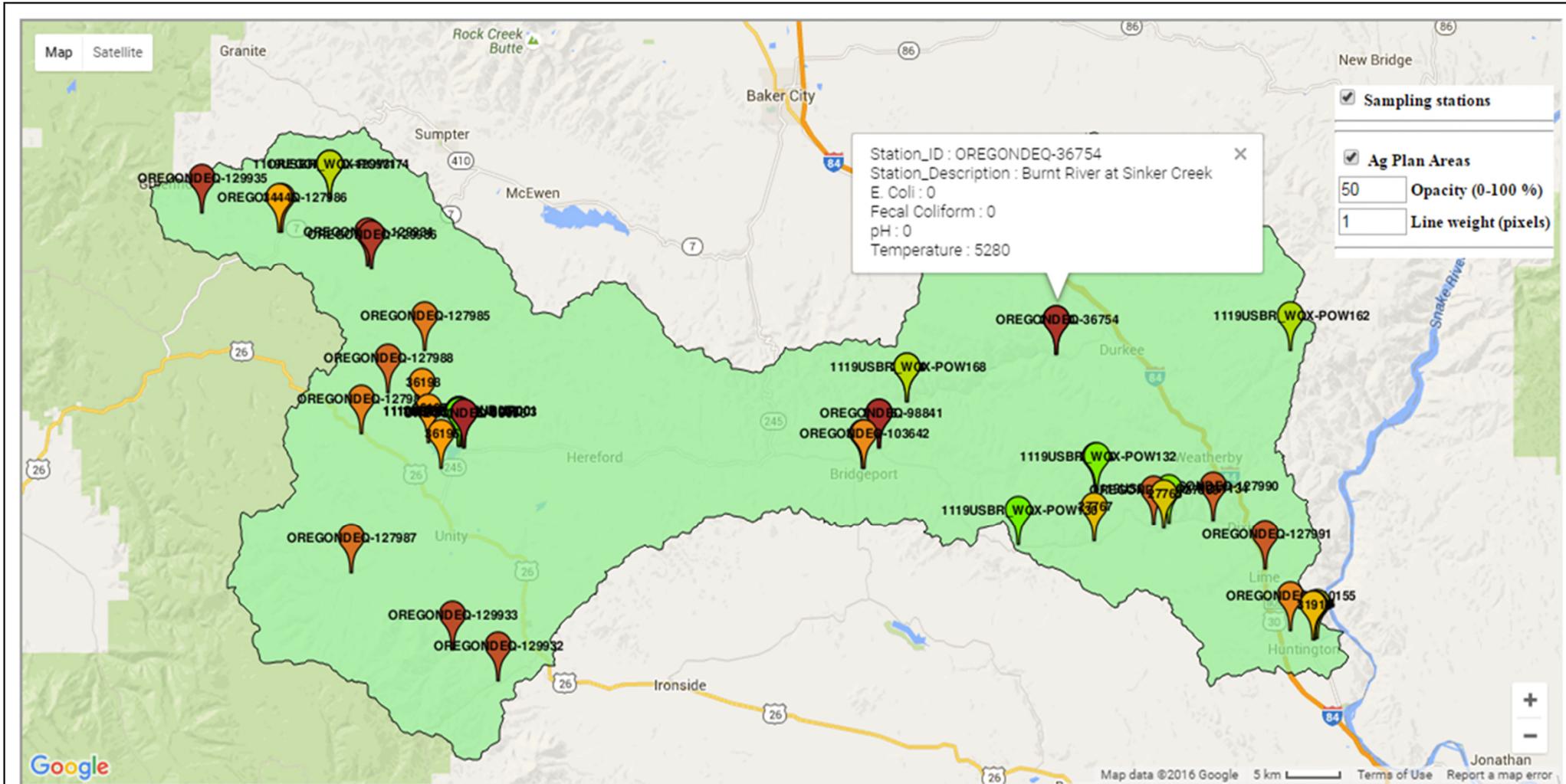
You just submitted Burnt River Plan Area Query for Temperature, pH, Bacteria from 2006-04-05 to 2016-04-05

 Download the data

View map

	Analyte	Stations	Results
1	E. Coli	16	502
2	Fecal Coliform	4	12
3	pH	23	7376
4	Temperature	43	94621

Search For Data



Analyze

Format

- Each database has different field names
- Built function to combine results from each database into a single table

Analyze

Look for trend

- Uses Seasonal Kendall trend analysis from wq package
- Apply to each station for each parameter

Analyze

Compare to Water Quality Standard

- Many of Oregon's standards are geo-specific (e.g. Temperature and pH)
- Data are gathered from many different sources
- Station location information may not be accurate
- User is able to specify geo-specific criteria to apply to the station for analysis



Analyze

Compare to Water Quality Standard

- Temperature standard is based on 7 Day Average Daily Maximum
- E. Coli and Enterococcus are based on 30-day Geometric Mean

Display

- Provide tabular summaries of formatted data
- Create interactive charts using ggplot2 package
- Provide zooming to specific time periods in the charts
- Provide download of chart to insert into report
- Construct responsive table of exceedances based on changes to selected water quality standard

Display

Oregon Water Quality Status and Trend Beta Version 2.0

Data Query

Review Data

Plot Status and Trend

Select Review table to view:

- Data in tabular format ▲
- Parameter results by station
- Data in tabular format
- WQ Limited Waters within Ag Area
- QA - Summary by organization
- QA - Result values modified
- QA - Data removal information
- QA - Unique comment values

Show entries

Search:

Client ▲	Analyte ◆	Station_ID ◆	Station_Description ◆	SampleType ◆	Result ◆	MRL ◆	Unit ◆
<input type="text"/>	<input type="text" value="A"/>	<input type="text" value="AI"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
1 Ambient Water Quality Monitoring - DEQ	E. Coli	11494	Burnt River at Snake River Road (Huntington)	Grab Sample::GS	22.0	1.0	MPN/100 mL
2 Ambient Water Quality Monitoring - DEQ	pH	11494	Burnt River at Snake River Road (Huntington)	Grab Sample::GS	8.5	0.1	pH Units
3 Ambient Water Quality Monitoring - DEQ	Temperature	11494	Burnt River at Snake River Road (Huntington)	Grab Sample::GS	8.3	0.0	°C

Display

Select station to evaluate:

12187 - Youngs River at Youngs River Loop Road

Select parameter to evaluate:

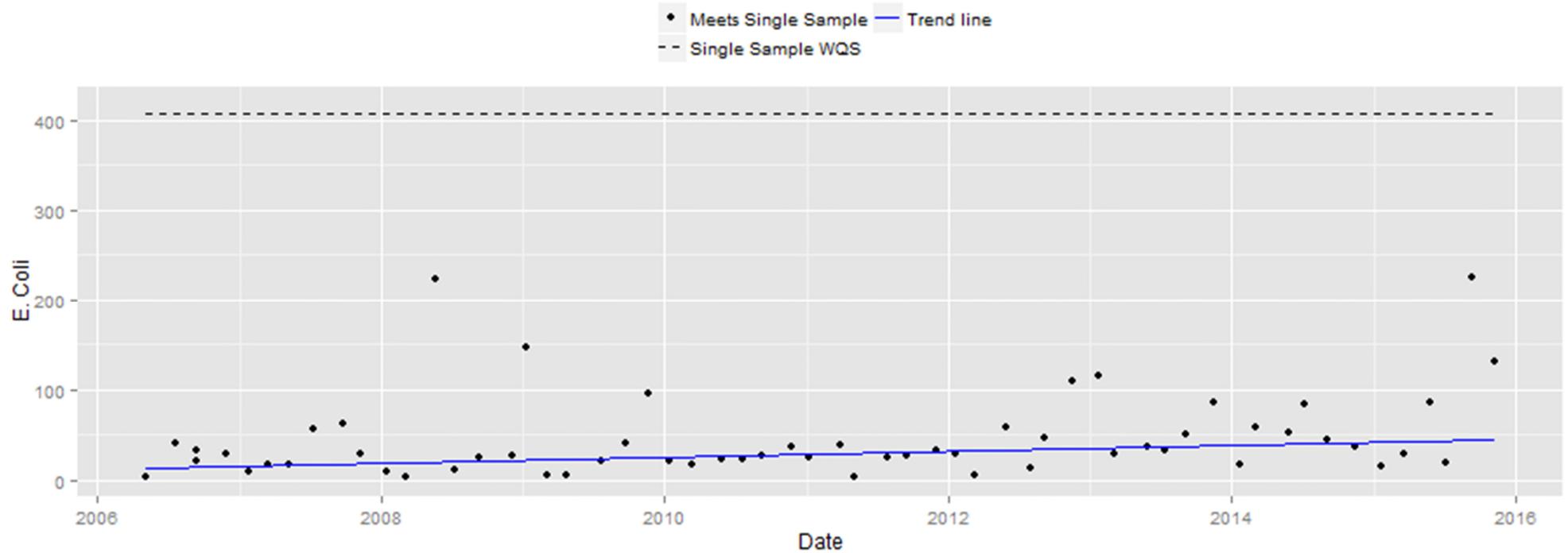
E. Coli

Plot data with log scale

Plot Seasonal Kendall trend line (Note: May not be significant)

Youngs River at Youngs River Loop Road, ID = 12187

p value = 0.003, 99% Significance Level, slope = 3.36, n = 59



Display

Data Query

Review Data

Plot Status and Trend

Select station to evaluate:

11434 - Clatskanie River at Hwy 30 (Clatskanie)

Select parameter to evaluate:

E. Coli

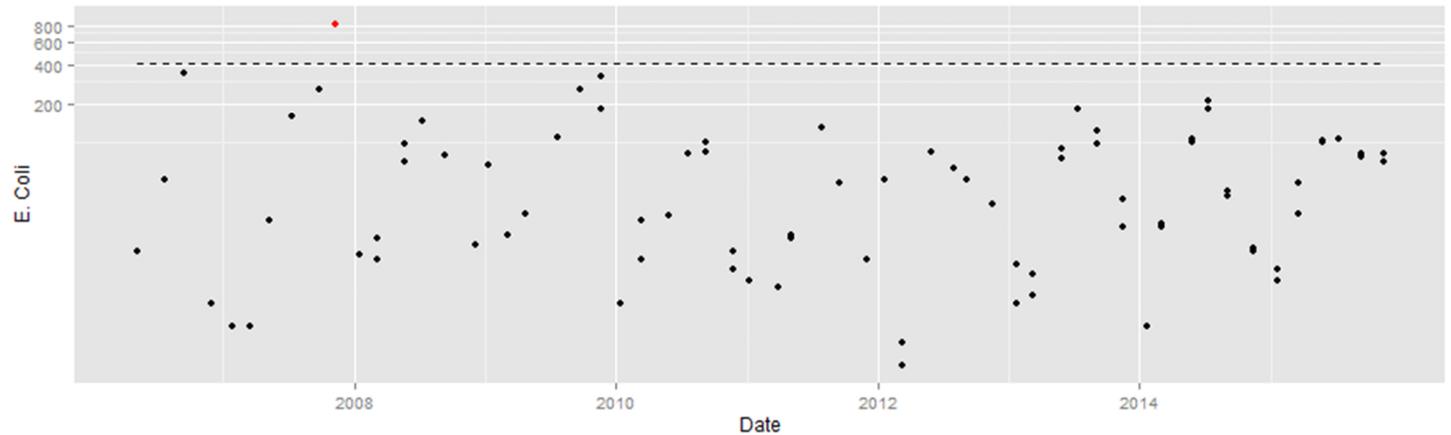
Plot data with log scale

Plot Seasonal Kendall trend line (Note: May not be significant)

Clatskanie River at Hwy 30 (Clatskanie), ID = 11434

p value = 0.384, Not Significant, slope = 0.75, n = 83

• Exceeds Single Sample - - Single Sample WQS
• Meets Single Sample



Save plot

Show 10 entries

Search:

	Station_ID	Station_Description	Sample	Obs	Exceedances	Percent_Exceed
1	11434	Clatskanie River at Hwy 30 (Clatskanie)	Single sample	83	1	1.204819
2	11434	Clatskanie River at Hwy 30 (Clatskanie)	Geomean	0	0	

Showing 1 to 2 of 2 entries

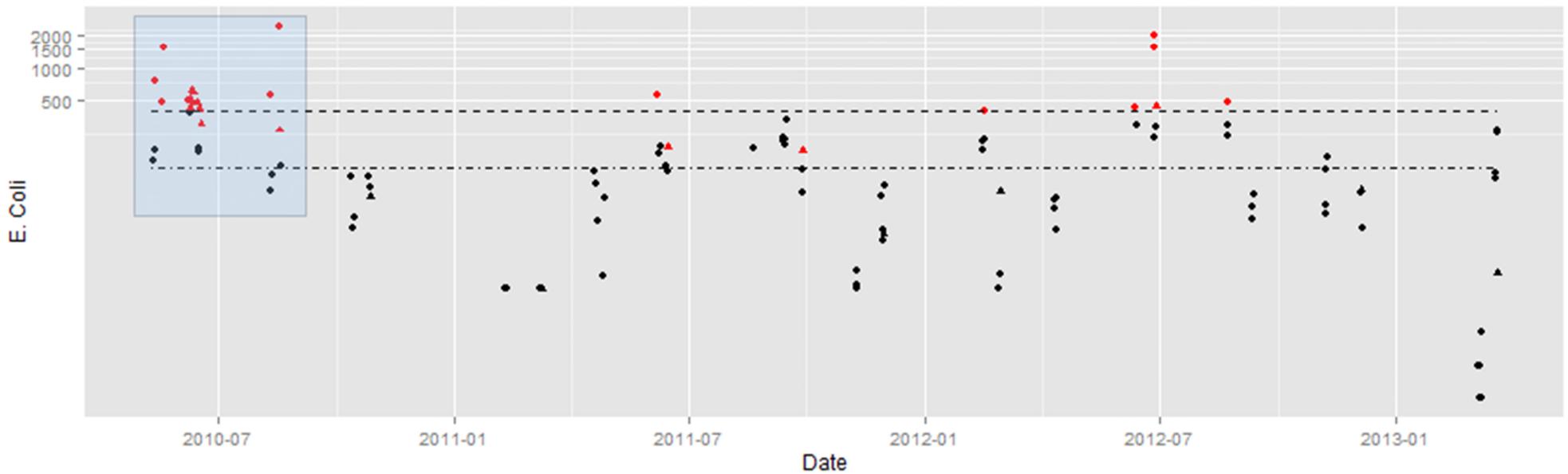
Previous 1 Next

Display

Burnt River at Clarks Creek bridge, ID = 34256

p value = 1, Not Significant, slope = 37.17, n = 89

- Exceeds Single Sample
- Exceeds Geometric Mean
- Meets Single Sample
- Meets Geometric Mean
- Geometric Mean WQS
- Single Sample WQS



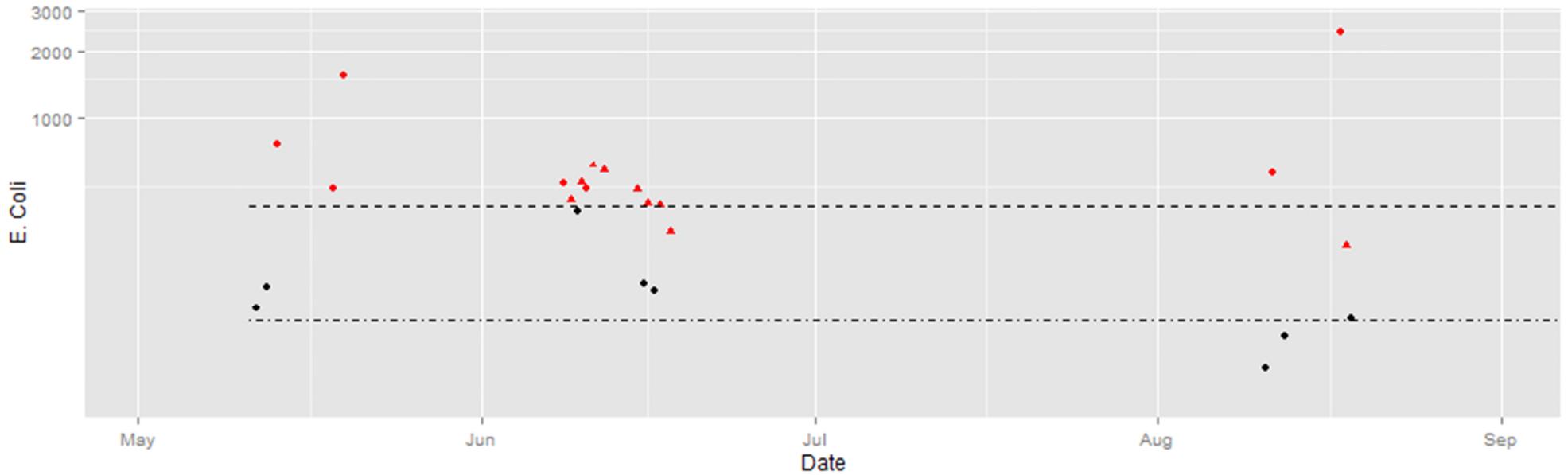
Save plot

Display

Burnt River at Clarks Creek bridge, ID = 34256

p value = 1, Not Significant, slope = 37.17, n = 89

- Exceeds Single Sample
- ▲ Exceeds Geometric Mean
- ◆ Meets Single Sample
- ▲ Meets Geometric Mean
- Geometric Mean WQS
- Single Sample WQS



Save plot

Display

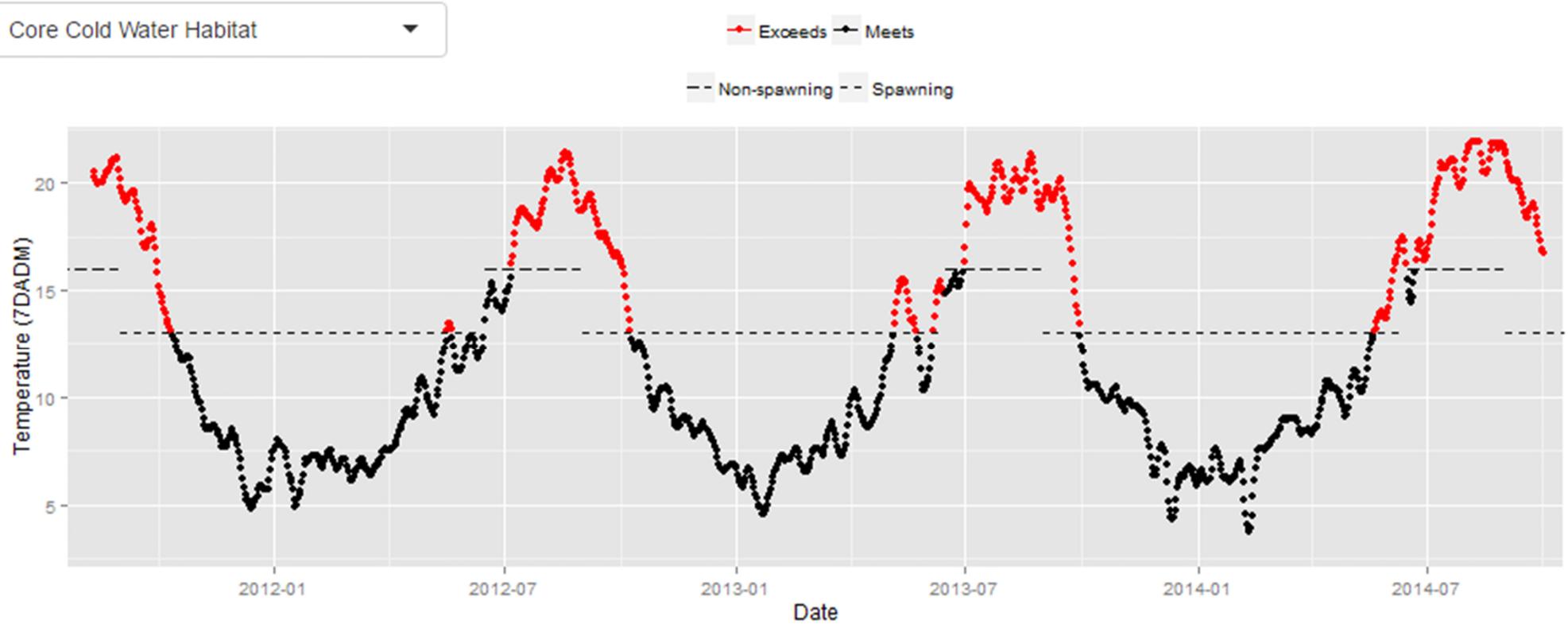
Select applicable spawning time period:

September 1-June 15

Select applicable beneficial fish use:

Core Cold Water Habitat

WILSON RIVER NEAR TILLAMOOK, OR, ID = USGS-14301500



Display

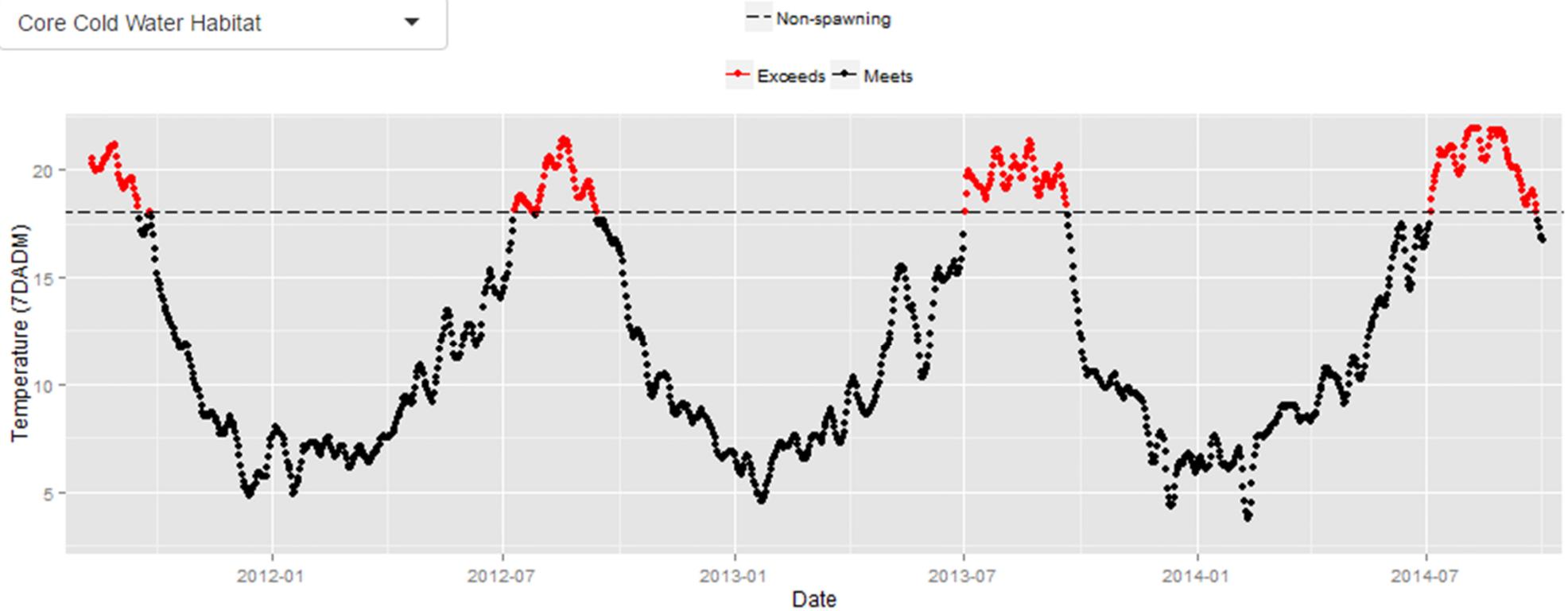
Select applicable spawning time period:

No spawning

Select applicable beneficial fish use:

Core Cold Water Habitat

WILSON RIVER NEAR TILLAMOOK, OR, ID = USGS-14301500



Summary

This R based web application is built using the Shiny package in R Studio and:

- Uses database queries to search for data
- R code to facilitate streamlined analysis
- Includes interactive tables and charts

If you want to see this

```
96
97
98 if ('Water Quality Portal' %in% input$db) {
99   incProgress(1/10, detail = 'Querying the water quality Portal')
100   prog <- prog + 1/10
101   wqpData <- tryCatch(wqpQuery(planArea = input$select,
102                               HUCList = HUCList,
103                               inParms = input$parms,
104                               luParms = parms,
105                               startDate = input$dates[1],
106                               endDate = input$dates[2]),
107                       error = function(err) {err <- geterrmessage()})
108
109 if (any(c('Temperature', 'pH') %in% input$parms)) {
110   incProgress(1/10, detail = 'Querying NWIS continuous data')
111   prog <- prog + 1/10
112   nwisData <- tryCatch(nwisQuery(planArea = input$select,
113                                  HUCList = HUCList,
114                                  inParms = input$parms,
115                                  startDate = input$dates[1],
116                                  endDate = input$dates[2]),
117                          error = function(err) {err <- geterrmessage()})
118 }
119
120 if (is.null(wqpData) & is.null(nwisData)) {
121   wqp_message <- 'Your query returned no results from the water quality Portal.'
122 } else if (!is.data.frame(wqpData) & !is.null(wqpData)) {
123   if (grepl("307", wqpData)) {
124     wqp_message <- 'Water Quality Portal is busy. Please try again in a few minutes.'
125   }
126 }
127
128 if ('DEQ' %in% input$db) {
129   incProgress(1/10, detail = 'Querying the LASAR database')
130   prog <- prog + 1/10
131
132   lasarData <- lasarQuery(planArea = input$select,
133                           HUCList = HUCList,
134                           inParms = input$parms,
135                           startDate = input$dates[1],
136                           endDate = input$dates[2])
137
138   odbcCloseAll()
139   if (nrow(lasarData) == 0) lasarData <- NULL
140
141   incProgress(1/10, detail = 'Querying the Element database')
142   prog <- prog + 1/10
143
144   elmData <- elementQuery(planArea = input$select,
145                            HUCList = HUCList,
146                            inParms = input$parms,
147                            startDate = input$dates[1],
148                            endDate = input$dates[2])
149
150   odbcCloseAll()
151   if (nrow(elmData) == 0) elmData <- NULL
152 }
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Go To This Project's Repository

<https://github.com/petertbryant/StatusAndTrends>

The screenshot shows the GitHub repository page for 'petertbryant / StatusAndTrends'. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. Below that, the repository name is displayed along with 'Unwatch', 'Star' (0), and 'Fork' (0) buttons. A navigation bar includes 'Code', 'Issues' (0), 'Pull requests' (0), 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The main content area features a header with '55 commits', '2 branches', '0 releases', and '1 contributor'. Below this is a toolbar with 'New pull request', 'New file', 'Upload files', 'Find file', 'SSH', and 'Download ZIP' options. A commit history table follows, listing recent updates by petertbryant. The 'README.md' file is selected and expanded, showing the title 'StatusAndTrends' and a description: 'Code to compile, clean, format, associate to standard and evaluate for status and trends'.

Branch	Commits	Branches	Releases	Contributors
master	55	2	0	1

File	Description	Time
GIS	Building up plots in ggplot2 to make them work with brushing feature.	26 days ago
app	Updated methods for generating exceedances df for temperature. correc...	4 days ago
.gitignore	Let's start with WQP and LASAR queries. These queries are from the re...	a year ago
ManualShinyChecking.R	Fixed temperature plotting to accomodate all possible combinations of...	4 days ago
README.md	Initial commit	a year ago
StatusAndTrends.Rproj	Let's start with WQP and LASAR queries. These queries are from the re...	a year ago
preprocess_303d_extract.R	Massive overhaul to update and simplify and make more reliable server...	26 days ago
textmining.R	Added testing folder to build up additional functionality while prese...	7 months ago

StatusAndTrends

Code to compile, clean, format, associate to standard and evaluate for status and trends

Thanks to

- Oregon DEQ Laboratory Water Quality Monitoring Group and Steve Hanson, Wade Peerman, Michael Tichenor and Lori Pillsbury
- John Paul Schmit with the National Park Service whose Shiny app provided inspiration and whose Github repository helped get me started

Questions?

Contact info:

bryant.peter@deg.state.or.us

GitHub

- Github is a website where you share your R code with others online
- Github uses the version control software git
- Version control software keeps track of the changes you make to your code