

R Notebook

```
install.packages(c("tidymodels","modeltime","dplyr","timetk", "ggplot2", "glmnet", "tidyquant", "magrittr"))
```

```
## Installing packages into '/home/provak/R/x86_64-pc-linux-gnu-library/4.3'  
## (as 'lib' is unspecified)
```

```
library(tidymodels)
```

```
## -- Attaching packages ----- tidymodels 1.1.1 --
```

```
## v broom      1.0.5    v recipes      1.0.9  
## v dials      1.2.0    v rsample      1.2.0  
## v dplyr      1.1.4    v tibble       3.2.1  
## v ggplot2    3.4.4    v tidyr        1.3.0  
## v infer      1.0.5    v tune         1.1.2  
## v modeldata  1.2.0    v workflows    1.1.3  
## v parsnip    1.1.1    v workflowsets 1.0.1  
## v purrr      1.0.2    v yardstick    1.2.0
```

```
## -- Conflicts ----- tidymodels_conflicts() --
```

```
## x purrr::discard() masks scales::discard()  
## x dplyr::filter()  masks stats::filter()  
## x dplyr::lag()     masks stats::lag()  
## x recipes::step() masks stats::step()  
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```
library(dplyr)  
library(timetk)  
library(magrittr)
```

```
##
```

```
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:tidyr':
```

```
##
```

```
##      extract
```

```
## The following object is masked from 'package:purrr':
```

```
##
```

```
##      set_names
```

```
library(workflows)  
library(parsnip)  
library(recipes)  
library(yardstick)  
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
```

```
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
```

```

##
##   expand, pack, unpack
## Loaded glmnet 4.1-8
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v readr   2.1.4      v forcats 1.0.0
## v stringr 1.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x readr::col_factor() masks scales::col_factor()
## x purrr::discard()     masks scales::discard()
## x Matrix::expand()     masks tidyr::expand()
## x magrittr::extract()  masks tidyr::extract()
## x dplyr::filter()      masks stats::filter()
## x stringr::fixed()     masks recipes::fixed()
## x dplyr::lag()          masks stats::lag()
## x Matrix::pack()       masks tidyr::pack()
## x magrittr::set_names() masks purrr::set_names()
## x readr::spec()        masks yardstick::spec()
## x Matrix::unpack()     masks tidyr::unpack()
library(tidyquant)

## Loading required package: lubridate

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##   date, intersect, setdiff, union

## Loading required package: PerformanceAnalytics

## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric

##
## ##### Warning from 'xts' package #####
## #
## # The dplyr lag() function breaks how base R's lag() function is supposed to #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or #
## # source() into this session won't work correctly. #
## #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop #
## # dplyr from breaking base R's lag() function. #
## #
## # Code in packages is not affected. It's protected by R's namespace mechanism #

```

```
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning. #
## #
## #####
##
## Attaching package: 'xts'
##
## The following objects are masked from 'package:dplyr':
##
##   first, last
##
## Attaching package: 'PerformanceAnalytics'
##
## The following object is masked from 'package:graphics':
##
##   legend
##
## Loading required package: quantmod
## Loading required package: TTR
##
## Attaching package: 'TTR'
##
## The following object is masked from 'package:dials':
##
##   momentum
##
## Registered S3 method overwritten by 'quantmod':
##   method      from
##   as.zoo.data.frame zoo
```

```
library(timetk)
library(kknn)
```

```
BC <- read.csv("BorderCrossingsFromUKR.csv") %>% select(timeseries.data_date, timeseries.individuals) %>%
  BC$date = as.Date(BC$date, format = '%Y-%m-%d')
BC
```

```
##           date people
## 1  2022-02-24   4844
## 2  2022-02-25   6941
## 3  2022-02-26  11995
## 4  2022-02-27  15881
## 5  2022-02-28  13035
## 6  2022-03-01  12620
## 7  2022-03-02  12466
## 8  2022-03-03  11242
## 9  2022-03-04  10784
## 10 2022-03-05  12650
## 11 2022-03-06  13496
## 12 2022-03-07  13345
## 13 2022-03-08  12442
## 14 2022-03-09  12359
## 15 2022-03-10  11316
## 16 2022-03-11  10664
## 17 2022-03-12  10363
## 18 2022-03-13   9957
## 19 2022-03-14   8495
```

##	20	2022-03-15	7804
##	21	2022-03-16	8099
##	22	2022-03-17	7243
##	23	2022-03-18	5175
##	24	2022-03-19	5464
##	25	2022-03-20	4409
##	26	2022-03-21	3305
##	27	2022-03-22	3370
##	28	2022-03-23	3324
##	29	2022-03-24	3614
##	30	2022-03-25	3673
##	31	2022-03-26	4215
##	32	2022-03-27	3337
##	33	2022-03-28	2741
##	34	2022-03-29	2860
##	35	2022-03-30	2817
##	36	2022-03-31	2740
##	37	2022-04-01	2747
##	38	2022-04-02	3199
##	39	2022-04-03	3121
##	40	2022-04-04	1991
##	41	2022-04-05	2274
##	42	2022-04-06	2566
##	43	2022-04-07	2789
##	44	2022-04-08	3118
##	45	2022-04-09	3595
##	46	2022-04-10	3296
##	47	2022-04-11	2465
##	48	2022-04-12	2774
##	49	2022-04-13	3224
##	50	2022-04-14	3353
##	51	2022-04-15	3110
##	52	2022-04-16	2536
##	53	2022-04-17	2068
##	54	2022-04-18	2369
##	55	2022-04-19	3133
##	56	2022-04-20	3362
##	57	2022-04-21	3111
##	58	2022-04-22	3021
##	59	2022-04-23	2043
##	60	2022-04-24	1264
##	61	2022-04-25	1967
##	62	2022-04-26	2877
##	63	2022-04-27	3482
##	64	2022-04-28	3964
##	65	2022-04-29	4014
##	66	2022-04-30	4218
##	67	2022-05-01	3311
##	68	2022-05-02	2577
##	69	2022-05-03	3260
##	70	2022-05-04	2998
##	71	2022-05-05	3310
##	72	2022-05-06	3308
##	73	2022-05-07	4084

##	74	2022-05-08	3348
##	75	2022-05-09	2131
##	76	2022-05-10	2370
##	77	2022-05-11	2694
##	78	2022-05-12	2818
##	79	2022-05-13	3057
##	80	2022-05-14	3510
##	81	2022-05-15	2750
##	82	2022-05-16	2365
##	83	2022-05-17	2578
##	84	2022-05-18	3100
##	85	2022-05-19	2797
##	86	2022-05-20	3158
##	87	2022-05-21	3660
##	88	2022-05-22	2996
##	89	2022-05-23	2052
##	90	2022-05-24	2387
##	91	2022-05-25	2754
##	92	2022-05-26	2341
##	93	2022-05-27	3111
##	94	2022-05-28	3400
##	95	2022-05-29	2803
##	96	2022-05-30	2377
##	97	2022-05-31	2723
##	98	2022-06-01	2758
##	99	2022-06-02	2400
##	100	2022-06-03	2577
##	101	2022-06-04	3354
##	102	2022-06-05	2945
##	103	2022-06-06	2180
##	104	2022-06-07	2183
##	105	2022-06-08	2492
##	106	2022-06-09	2745
##	107	2022-06-10	3110
##	108	2022-06-11	3516
##	109	2022-06-12	2570
##	110	2022-06-13	2241
##	111	2022-06-14	2713
##	112	2022-06-15	3230
##	113	2022-06-16	2736
##	114	2022-06-17	3419
##	115	2022-06-18	3728
##	116	2022-06-19	3262
##	117	2022-06-20	2446
##	118	2022-06-21	2751
##	119	2022-06-22	3107
##	120	2022-06-23	3177
##	121	2022-06-24	3487
##	122	2022-06-25	3953
##	123	2022-06-26	3100
##	124	2022-06-27	2539
##	125	2022-06-28	3227
##	126	2022-06-29	3341
##	127	2022-06-30	3445

##	128	2022-07-01	3134
##	129	2022-07-02	3382
##	130	2022-07-03	3142
##	131	2022-07-04	2447
##	132	2022-07-05	2601
##	133	2022-07-06	3293
##	134	2022-07-07	2498
##	135	2022-07-08	2983
##	136	2022-07-09	3263
##	137	2022-07-10	2900
##	138	2022-07-11	2160
##	139	2022-07-12	2348
##	140	2022-07-13	3144
##	141	2022-07-14	3222
##	142	2022-07-15	3110
##	143	2022-07-16	3553
##	144	2022-07-17	3303
##	145	2022-07-18	2181
##	146	2022-07-19	2541
##	147	2022-07-20	2762
##	148	2022-07-21	2470
##	149	2022-07-22	2321
##	150	2022-07-23	2976
##	151	2022-07-24	2882
##	152	2022-07-25	1941
##	153	2022-07-26	2002
##	154	2022-07-27	2589
##	155	2022-07-28	2907
##	156	2022-07-29	3347
##	157	2022-07-30	3551
##	158	2022-07-31	3504
##	159	2022-08-01	2366
##	160	2022-08-02	2693
##	161	2022-08-03	2791
##	162	2022-08-04	2639
##	163	2022-08-05	2695
##	164	2022-08-06	3581
##	165	2022-08-07	3271
##	166	2022-08-08	2307
##	167	2022-08-09	2399
##	168	2022-08-10	3177
##	169	2022-08-11	2995
##	170	2022-08-12	3111
##	171	2022-08-13	3593
##	172	2022-08-14	3730
##	173	2022-08-15	2567
##	174	2022-08-16	2969
##	175	2022-08-17	3126
##	176	2022-08-18	3257
##	177	2022-08-19	3448
##	178	2022-08-20	3721
##	179	2022-08-21	3466
##	180	2022-08-22	3018
##	181	2022-08-23	3215

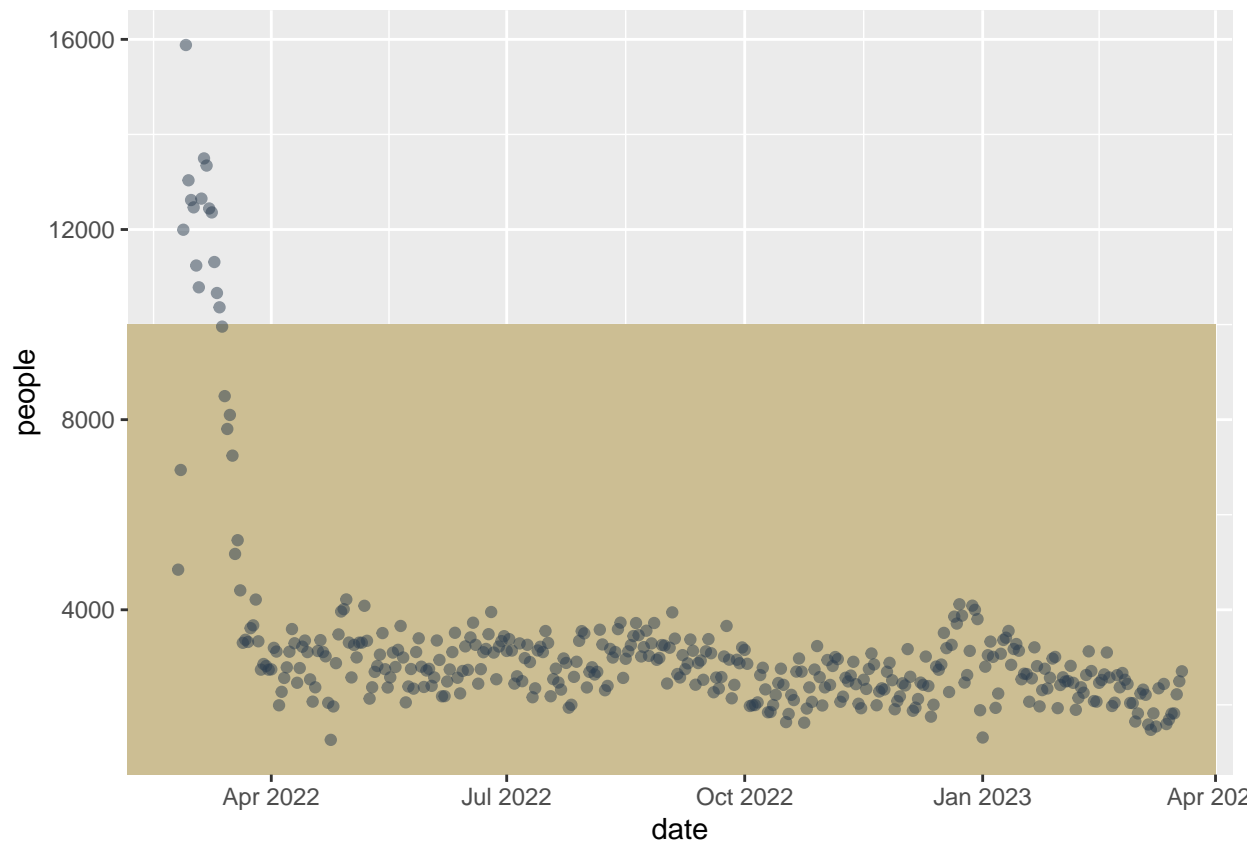
##	182	2022-08-24	3560
##	183	2022-08-25	3030
##	184	2022-08-26	3293
##	185	2022-08-27	3722
##	186	2022-08-28	2948
##	187	2022-08-29	2987
##	188	2022-08-30	3259
##	189	2022-08-31	3248
##	190	2022-09-01	2449
##	191	2022-09-02	3202
##	192	2022-09-03	3946
##	193	2022-09-04	3394
##	194	2022-09-05	2649
##	195	2022-09-06	2576
##	196	2022-09-07	3046
##	197	2022-09-08	2750
##	198	2022-09-09	2872
##	199	2022-09-10	3375
##	200	2022-09-11	3138
##	201	2022-09-12	2424
##	202	2022-09-13	2879
##	203	2022-09-14	2933
##	204	2022-09-15	2528
##	205	2022-09-16	3127
##	206	2022-09-17	3383
##	207	2022-09-18	3086
##	208	2022-09-19	2265
##	209	2022-09-20	2575
##	210	2022-09-21	2345
##	211	2022-09-22	2585
##	212	2022-09-23	3018
##	213	2022-09-24	3660
##	214	2022-09-25	2951
##	215	2022-09-26	2138
##	216	2022-09-27	2421
##	217	2022-09-28	2949
##	218	2022-09-29	2876
##	219	2022-09-30	3208
##	220	2022-10-01	3156
##	221	2022-10-02	2864
##	222	2022-10-03	1976
##	223	2022-10-04	1993
##	224	2022-10-05	1995
##	225	2022-10-06	2052
##	226	2022-10-07	2627
##	227	2022-10-08	2781
##	228	2022-10-09	2326
##	229	2022-10-10	1844
##	230	2022-10-11	1850
##	231	2022-10-12	1996
##	232	2022-10-13	2210
##	233	2022-10-14	2460
##	234	2022-10-15	2761
##	235	2022-10-16	2421

##	236	2022-10-17	1637
##	237	2022-10-18	1809
##	238	2022-10-19	2211
##	239	2022-10-20	2100
##	240	2022-10-21	2700
##	241	2022-10-22	2976
##	242	2022-10-23	2699
##	243	2022-10-24	1625
##	244	2022-10-25	1922
##	245	2022-10-26	2368
##	246	2022-10-27	2064
##	247	2022-10-28	2743
##	248	2022-10-29	3237
##	249	2022-10-30	2584
##	250	2022-10-31	1987
##	251	2022-11-01	2367
##	252	2022-11-02	2944
##	253	2022-11-03	2420
##	254	2022-11-04	2818
##	255	2022-11-05	3006
##	256	2022-11-06	2962
##	257	2022-11-07	2065
##	258	2022-11-08	2173
##	259	2022-11-09	2571
##	260	2022-11-10	2493
##	261	2022-11-11	2616
##	262	2022-11-12	2906
##	263	2022-11-13	2435
##	264	2022-11-14	2024
##	265	2022-11-15	1929
##	266	2022-11-16	2529
##	267	2022-11-17	2335
##	268	2022-11-18	2755
##	269	2022-11-19	3081
##	270	2022-11-20	2851
##	271	2022-11-21	1992
##	272	2022-11-22	2281
##	273	2022-11-23	2349
##	274	2022-11-24	2323
##	275	2022-11-25	2697
##	276	2022-11-26	2884
##	277	2022-11-27	2518
##	278	2022-11-28	1907
##	279	2022-11-29	2079
##	280	2022-11-30	2178
##	281	2022-12-01	2454
##	282	2022-12-02	2408
##	283	2022-12-03	3173
##	284	2022-12-04	2591
##	285	2022-12-05	1876
##	286	2022-12-06	1942
##	287	2022-12-07	2123
##	288	2022-12-08	2469
##	289	2022-12-09	2421

##	290	2022-12-10	3016
##	291	2022-12-11	2396
##	292	2022-12-12	1752
##	293	2022-12-13	2003
##	294	2022-12-14	2806
##	295	2022-12-15	2739
##	296	2022-12-16	2848
##	297	2022-12-17	3514
##	298	2022-12-18	3193
##	299	2022-12-19	2270
##	300	2022-12-20	3262
##	301	2022-12-21	3856
##	302	2022-12-22	3710
##	303	2022-12-23	4116
##	304	2022-12-24	3877
##	305	2022-12-25	2468
##	306	2022-12-26	2627
##	307	2022-12-27	3135
##	308	2022-12-28	4087
##	309	2022-12-29	3999
##	310	2022-12-30	3804
##	311	2022-12-31	1887
##	312	2023-01-01	1313
##	313	2023-01-02	2803
##	314	2023-01-03	3045
##	315	2023-01-04	3331
##	316	2023-01-05	3010
##	317	2023-01-06	1937
##	318	2023-01-07	2238
##	319	2023-01-08	3075
##	320	2023-01-09	3365
##	321	2023-01-10	3413
##	322	2023-01-11	3555
##	323	2023-01-12	2842
##	324	2023-01-13	3175
##	325	2023-01-14	3280
##	326	2023-01-15	3139
##	327	2023-01-16	2538
##	328	2023-01-17	2656
##	329	2023-01-18	2649
##	330	2023-01-19	2066
##	331	2023-01-20	2557
##	332	2023-01-21	3210
##	333	2023-01-22	2815
##	334	2023-01-23	1968
##	335	2023-01-24	2310
##	336	2023-01-25	2760
##	337	2023-01-26	2344
##	338	2023-01-27	2571
##	339	2023-01-28	2974
##	340	2023-01-29	3009
##	341	2023-01-30	1931
##	342	2023-01-31	2418
##	343	2023-02-01	2580

```
## 344 2023-02-02 2492
## 345 2023-02-03 2500
## 346 2023-02-04 2820
## 347 2023-02-05 2463
## 348 2023-02-06 1896
## 349 2023-02-07 2147
## 350 2023-02-08 2370
## 351 2023-02-09 2255
## 352 2023-02-10 2629
## 353 2023-02-11 3126
## 354 2023-02-12 2712
## 355 2023-02-13 2080
## 356 2023-02-14 2067
## 357 2023-02-15 2464
## 358 2023-02-16 2523
## 359 2023-02-17 2641
## 360 2023-02-18 3102
## 361 2023-02-19 2576
## 362 2023-02-20 1976
## 363 2023-02-21 2046
## 364 2023-02-22 2625
## 365 2023-02-23 2369
## 366 2023-02-24 2670
## 367 2023-02-25 2528
## 368 2023-02-26 2439
## 369 2023-02-27 2040
## 370 2023-02-28 2036
## 371 2023-03-01 1649
## 372 2023-03-02 1820
## 373 2023-03-03 2239
## 374 2023-03-04 2320
## 375 2023-03-05 2212
## 376 2023-03-06 1587
## 377 2023-03-07 1475
## 378 2023-03-08 1821
## 379 2023-03-09 1545
## 380 2023-03-10 2348
## 381 2023-03-12 2440
## 382 2023-03-13 1596
## 383 2023-03-14 1691
## 384 2023-03-15 1814
## 385 2023-03-16 1821
## 386 2023-03-17 2222
## 387 2023-03-18 2496
## 388 2023-03-19 2706
```

```
BC %>%
  ggplot(aes(x = date, y = people)) +
    geom_rect(xmin = as.numeric(ymd("2022-02-01")),
              xmax = as.numeric(ymd("2023-04-01")),
              ymin = 0, ymax = 10000,
              fill = palette_light()[[4]], alpha = 0.01) +
    geom_point(alpha = 0.5, color = palette_light()[[1]])
```



```
train_BC <- BC %>% filter(date < ymd("2023-01-01") & date >= ymd("2022-04-01"))
test_BC <- BC %>% filter(date >= ymd("2023-01-01"))
```

```
train_BC
```

##	date	people
## 1	2022-04-01	2747
## 2	2022-04-02	3199
## 3	2022-04-03	3121
## 4	2022-04-04	1991
## 5	2022-04-05	2274
## 6	2022-04-06	2566
## 7	2022-04-07	2789
## 8	2022-04-08	3118
## 9	2022-04-09	3595
## 10	2022-04-10	3296
## 11	2022-04-11	2465
## 12	2022-04-12	2774
## 13	2022-04-13	3224
## 14	2022-04-14	3353
## 15	2022-04-15	3110
## 16	2022-04-16	2536
## 17	2022-04-17	2068
## 18	2022-04-18	2369
## 19	2022-04-19	3133
## 20	2022-04-20	3362
## 21	2022-04-21	3111

##	22	2022-04-22	3021
##	23	2022-04-23	2043
##	24	2022-04-24	1264
##	25	2022-04-25	1967
##	26	2022-04-26	2877
##	27	2022-04-27	3482
##	28	2022-04-28	3964
##	29	2022-04-29	4014
##	30	2022-04-30	4218
##	31	2022-05-01	3311
##	32	2022-05-02	2577
##	33	2022-05-03	3260
##	34	2022-05-04	2998
##	35	2022-05-05	3310
##	36	2022-05-06	3308
##	37	2022-05-07	4084
##	38	2022-05-08	3348
##	39	2022-05-09	2131
##	40	2022-05-10	2370
##	41	2022-05-11	2694
##	42	2022-05-12	2818
##	43	2022-05-13	3057
##	44	2022-05-14	3510
##	45	2022-05-15	2750
##	46	2022-05-16	2365
##	47	2022-05-17	2578
##	48	2022-05-18	3100
##	49	2022-05-19	2797
##	50	2022-05-20	3158
##	51	2022-05-21	3660
##	52	2022-05-22	2996
##	53	2022-05-23	2052
##	54	2022-05-24	2387
##	55	2022-05-25	2754
##	56	2022-05-26	2341
##	57	2022-05-27	3111
##	58	2022-05-28	3400
##	59	2022-05-29	2803
##	60	2022-05-30	2377
##	61	2022-05-31	2723
##	62	2022-06-01	2758
##	63	2022-06-02	2400
##	64	2022-06-03	2577
##	65	2022-06-04	3354
##	66	2022-06-05	2945
##	67	2022-06-06	2180
##	68	2022-06-07	2183
##	69	2022-06-08	2492
##	70	2022-06-09	2745
##	71	2022-06-10	3110
##	72	2022-06-11	3516
##	73	2022-06-12	2570
##	74	2022-06-13	2241
##	75	2022-06-14	2713

##	76	2022-06-15	3230
##	77	2022-06-16	2736
##	78	2022-06-17	3419
##	79	2022-06-18	3728
##	80	2022-06-19	3262
##	81	2022-06-20	2446
##	82	2022-06-21	2751
##	83	2022-06-22	3107
##	84	2022-06-23	3177
##	85	2022-06-24	3487
##	86	2022-06-25	3953
##	87	2022-06-26	3100
##	88	2022-06-27	2539
##	89	2022-06-28	3227
##	90	2022-06-29	3341
##	91	2022-06-30	3445
##	92	2022-07-01	3134
##	93	2022-07-02	3382
##	94	2022-07-03	3142
##	95	2022-07-04	2447
##	96	2022-07-05	2601
##	97	2022-07-06	3293
##	98	2022-07-07	2498
##	99	2022-07-08	2983
##	100	2022-07-09	3263
##	101	2022-07-10	2900
##	102	2022-07-11	2160
##	103	2022-07-12	2348
##	104	2022-07-13	3144
##	105	2022-07-14	3222
##	106	2022-07-15	3110
##	107	2022-07-16	3553
##	108	2022-07-17	3303
##	109	2022-07-18	2181
##	110	2022-07-19	2541
##	111	2022-07-20	2762
##	112	2022-07-21	2470
##	113	2022-07-22	2321
##	114	2022-07-23	2976
##	115	2022-07-24	2882
##	116	2022-07-25	1941
##	117	2022-07-26	2002
##	118	2022-07-27	2589
##	119	2022-07-28	2907
##	120	2022-07-29	3347
##	121	2022-07-30	3551
##	122	2022-07-31	3504
##	123	2022-08-01	2366
##	124	2022-08-02	2693
##	125	2022-08-03	2791
##	126	2022-08-04	2639
##	127	2022-08-05	2695
##	128	2022-08-06	3581
##	129	2022-08-07	3271

##	130	2022-08-08	2307
##	131	2022-08-09	2399
##	132	2022-08-10	3177
##	133	2022-08-11	2995
##	134	2022-08-12	3111
##	135	2022-08-13	3593
##	136	2022-08-14	3730
##	137	2022-08-15	2567
##	138	2022-08-16	2969
##	139	2022-08-17	3126
##	140	2022-08-18	3257
##	141	2022-08-19	3448
##	142	2022-08-20	3721
##	143	2022-08-21	3466
##	144	2022-08-22	3018
##	145	2022-08-23	3215
##	146	2022-08-24	3560
##	147	2022-08-25	3030
##	148	2022-08-26	3293
##	149	2022-08-27	3722
##	150	2022-08-28	2948
##	151	2022-08-29	2987
##	152	2022-08-30	3259
##	153	2022-08-31	3248
##	154	2022-09-01	2449
##	155	2022-09-02	3202
##	156	2022-09-03	3946
##	157	2022-09-04	3394
##	158	2022-09-05	2649
##	159	2022-09-06	2576
##	160	2022-09-07	3046
##	161	2022-09-08	2750
##	162	2022-09-09	2872
##	163	2022-09-10	3375
##	164	2022-09-11	3138
##	165	2022-09-12	2424
##	166	2022-09-13	2879
##	167	2022-09-14	2933
##	168	2022-09-15	2528
##	169	2022-09-16	3127
##	170	2022-09-17	3383
##	171	2022-09-18	3086
##	172	2022-09-19	2265
##	173	2022-09-20	2575
##	174	2022-09-21	2345
##	175	2022-09-22	2585
##	176	2022-09-23	3018
##	177	2022-09-24	3660
##	178	2022-09-25	2951
##	179	2022-09-26	2138
##	180	2022-09-27	2421
##	181	2022-09-28	2949
##	182	2022-09-29	2876
##	183	2022-09-30	3208

##	184	2022-10-01	3156
##	185	2022-10-02	2864
##	186	2022-10-03	1976
##	187	2022-10-04	1993
##	188	2022-10-05	1995
##	189	2022-10-06	2052
##	190	2022-10-07	2627
##	191	2022-10-08	2781
##	192	2022-10-09	2326
##	193	2022-10-10	1844
##	194	2022-10-11	1850
##	195	2022-10-12	1996
##	196	2022-10-13	2210
##	197	2022-10-14	2460
##	198	2022-10-15	2761
##	199	2022-10-16	2421
##	200	2022-10-17	1637
##	201	2022-10-18	1809
##	202	2022-10-19	2211
##	203	2022-10-20	2100
##	204	2022-10-21	2700
##	205	2022-10-22	2976
##	206	2022-10-23	2699
##	207	2022-10-24	1625
##	208	2022-10-25	1922
##	209	2022-10-26	2368
##	210	2022-10-27	2064
##	211	2022-10-28	2743
##	212	2022-10-29	3237
##	213	2022-10-30	2584
##	214	2022-10-31	1987
##	215	2022-11-01	2367
##	216	2022-11-02	2944
##	217	2022-11-03	2420
##	218	2022-11-04	2818
##	219	2022-11-05	3006
##	220	2022-11-06	2962
##	221	2022-11-07	2065
##	222	2022-11-08	2173
##	223	2022-11-09	2571
##	224	2022-11-10	2493
##	225	2022-11-11	2616
##	226	2022-11-12	2906
##	227	2022-11-13	2435
##	228	2022-11-14	2024
##	229	2022-11-15	1929
##	230	2022-11-16	2529
##	231	2022-11-17	2335
##	232	2022-11-18	2755
##	233	2022-11-19	3081
##	234	2022-11-20	2851
##	235	2022-11-21	1992
##	236	2022-11-22	2281
##	237	2022-11-23	2349

```
## 238 2022-11-24 2323
## 239 2022-11-25 2697
## 240 2022-11-26 2884
## 241 2022-11-27 2518
## 242 2022-11-28 1907
## 243 2022-11-29 2079
## 244 2022-11-30 2178
## 245 2022-12-01 2454
## 246 2022-12-02 2408
## 247 2022-12-03 3173
## 248 2022-12-04 2591
## 249 2022-12-05 1876
## 250 2022-12-06 1942
## 251 2022-12-07 2123
## 252 2022-12-08 2469
## 253 2022-12-09 2421
## 254 2022-12-10 3016
## 255 2022-12-11 2396
## 256 2022-12-12 1752
## 257 2022-12-13 2003
## 258 2022-12-14 2806
## 259 2022-12-15 2739
## 260 2022-12-16 2848
## 261 2022-12-17 3514
## 262 2022-12-18 3193
## 263 2022-12-19 2270
## 264 2022-12-20 3262
## 265 2022-12-21 3856
## 266 2022-12-22 3710
## 267 2022-12-23 4116
## 268 2022-12-24 3877
## 269 2022-12-25 2468
## 270 2022-12-26 2627
## 271 2022-12-27 3135
## 272 2022-12-28 4087
## 273 2022-12-29 3999
## 274 2022-12-30 3804
## 275 2022-12-31 1887
```

```
# Add time series signature
recipe_spec_timeseries <- recipe(people ~ date, data = train_BC) %>%
  step_timeseries_signature(date) %>% #adding time series signatures
  step_rm(date) %>%
  step_rm(contains("iso"),
    contains("minute"),
    contains("hour"),
    contains("am.pm"),
    contains("xts")) %>%
  step_zv(date_year)%>%
  step_dummy(contains("lbl"), one_hot = TRUE)
```

```
bake(prepare(recipe_spec_timeseries), new_data = train_BC)
```

```
## # A tibble: 275 x 36
##   people date_index.num date_half date_quarter date_month date_day date_second
```



```

##      <int>      <dbl>      <int>      <int>      <int>      <int>      <int>
## 1  2747      1648771200      1          2          4          1          0
## 2  3199      1648857600      1          2          4          2          0
## 3  3121      1648944000      1          2          4          3          0
## 4  1991      1649030400      1          2          4          4          0
## 5  2274      1649116800      1          2          4          5          0
## 6  2566      1649203200      1          2          4          6          0
## 7  2789      1649289600      1          2          4          7          0
## 8  3118      1649376000      1          2          4          8          0
## 9  3595      1649462400      1          2          4          9          0
## 10 3296      1649548800      1          2          4         10          0
## # i 265 more rows
## # i 29 more variables: date_wday <int>, date_mday <int>, date_qday <int>,
## #   date_yday <int>, date_mweek <int>, date_week <int>, date_week2 <int>,
## #   date_week3 <int>, date_week4 <int>, date_mday7 <int>,
## #   date_month.lbl_01 <dbl>, date_month.lbl_02 <dbl>, date_month.lbl_03 <dbl>,
## #   date_month.lbl_04 <dbl>, date_month.lbl_05 <dbl>, date_month.lbl_06 <dbl>,
## #   date_month.lbl_07 <dbl>, date_month.lbl_08 <dbl>, ...

model_spec_glmnet <- nearest_neighbor(neighbors = 3, weight_func = "gaussian") %>%
  # Using K-NN Regression
  set_mode("regression") %>%
  set_engine("knn")

workflow_glmnet <- workflow() %>%
  add_recipe(recipe_spec_timeseries) %>%
  add_model(model_spec_glmnet)

workflow_glmnet

## == Workflow ==
## Preprocessor: Recipe
## Model: nearest_neighbor()
##
## -- Preprocessor -----
## 5 Recipe Steps
##
## * step_timeseries_signature()
## * step_rm()
## * step_rm()
## * step_zv()
## * step_dummy()
##
## -- Model -----
## K-Nearest Neighbor Model Specification (regression)
##
## Main Arguments:
##   neighbors = 3
##   weight_func = gaussian
##
## Computational engine: knn

workflow_trained <- workflow_glmnet %>% fit(data = train_BC)

prediction_BC <- workflow_trained %>%

```

```

predict(test_BC) %>%
  bind_cols(test_BC)

prediction_BC

## # A tibble: 77 x 3
##   .pred date      people
##   <dbl> <date>    <int>
## 1 3259. 2023-01-01   1313
## 2 2232. 2023-01-02   2803
## 3 2633. 2023-01-03   3045
## 4 2752. 2023-01-04   3331
## 5 2972. 2023-01-05   3010
## 6 2964. 2023-01-06   1937
## 7 3633. 2023-01-07   2238
## 8 3075. 2023-01-08   3075
## 9 2156. 2023-01-09   3365
## 10 2402. 2023-01-10   3413
## # i 67 more rows

prediction_BC %>% metrics(people, .pred)

## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>      <dbl>
## 1 rmse    standard     738.
## 2 rsq     standard     0.0566
## 3 mae     standard     636.

idx <- BC %>% tk_index()
BC_sum <- idx %>% tk_get_timeseries_summary()

idx_future <- idx %>% tk_make_future_timeseries(length_out = 365)
future_BC <- tibble(date = idx_future)

future_BC

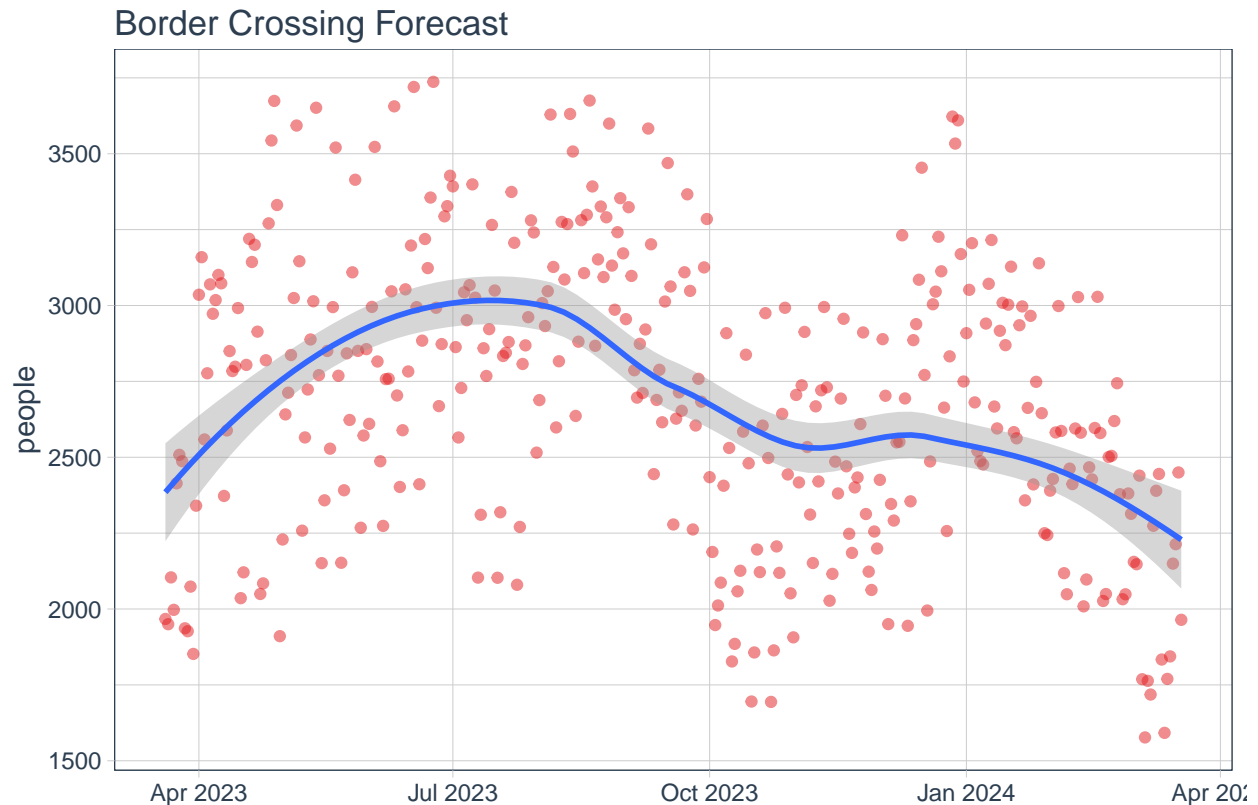
## # A tibble: 365 x 1
##   date
##   <date>
## 1 2023-03-20
## 2 2023-03-21
## 3 2023-03-22
## 4 2023-03-23
## 5 2023-03-24
## 6 2023-03-25
## 7 2023-03-26
## 8 2023-03-27
## 9 2023-03-28
## 10 2023-03-29
## # i 355 more rows

future_predictions_BC <- workflow_glmnet %>%
  fit(data = BC[BC$date >= "2022-04-01",]) %>%
  predict(future_BC) %>%
  bind_cols(future_BC)

```

```
BC %>%
  ggplot(aes(x = date, y = people)) +
    # future data
    geom_point(aes(x = date, y = .pred), data = future_predictions_BC,
               alpha = 0.5, color = palette_light()[[2]]) +
    geom_smooth(aes(x = date, y = .pred), data = future_predictions_BC,
                method = 'loess') +
    labs(title = "Border Crossing Forecast", x = "") +
    theme_tq()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
test_resid_sd <- prediction_BC %>%
  summarize(stdev = sd(people - .pred))
future_predictions_BC <- future_predictions_BC %>%
  mutate(
    lo.95 = .pred - 1.96 * test_resid_sd$stdev,
    lo.80 = .pred - 1.28 * test_resid_sd$stdev,
    hi.80 = .pred + 1.28 * test_resid_sd$stdev,
    hi.95 = .pred + 1.96 * test_resid_sd$stdev
  )
```

```
options(repr.plot.width=20, repr.plot.height=10)
BC %>%
  ggplot(aes(x = date, y = people)) +
    geom_point(alpha = 0, color = palette_light()[[1]]) +
    geom_line(alpha = 0.9)+
    geom_ribbon(aes(y = .pred, ymin = lo.95, ymax = hi.95 ),
               data = future_predictions_BC,
```

```

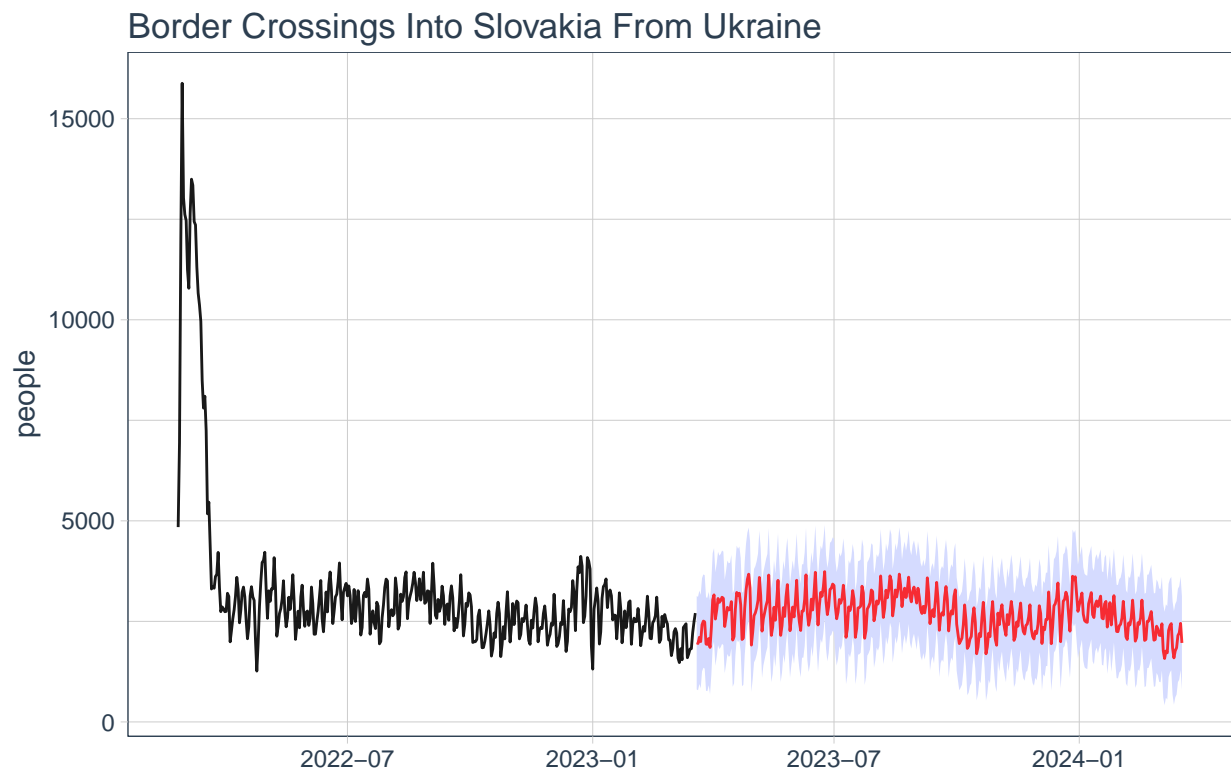
    fill = "#D5DBFF", color = NA, size = 0) +
# geom_ribbon(aes(y = .pred, ymin = lo.80, ymax = hi.80, fill = key),
#           data = future_predictions_BC,
#           fill = "#596DD5", color = NA, size = 0, alpha = 0.6) +
geom_point(aes(x = date, y = .pred), data = future_predictions_BC,
           alpha = 0, color = palette_light()[[2]]) +
geom_line(aes(x = date, y = .pred), data = future_predictions_BC,
          alpha = 0.8, color = "red") +
# geom_smooth(aes(x = date, y = .pred), data = future_predictions_BC,
#            method = 'loess', color = "red", alpha=0.1) +
labs(title = "Border Crossings Into Slovakia From Ukraine", x = "", caption='Shaded region indicates
theme_tq()

```

```

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```



Shaded region indicates 95% CI.

```
future_predictions_BC
```

```

## # A tibble: 365 x 6
##   .pred date      lo.95 lo.80 hi.80 hi.95
##   <dbl> <date>      <dbl> <dbl> <dbl> <dbl>
## 1 1967. 2023-03-20  807. 1209. 2725. 3128.
## 2 1950. 2023-03-21  789. 1192. 2708. 3111.
## 3 2104. 2023-03-22  944. 1346. 2862. 3265.

```

```
## 4 1997. 2023-03-23 837. 1239. 2755. 3158.  
## 5 2414. 2023-03-24 1254. 1656. 3172. 3575.  
## 6 2508. 2023-03-25 1347. 1750. 3266. 3669.  
## 7 2487. 2023-03-26 1326. 1729. 3245. 3648.  
## 8 1936. 2023-03-27 776. 1178. 2694. 3097.  
## 9 1927. 2023-03-28 766. 1169. 2685. 3087.  
## 10 2074. 2023-03-29 913. 1316. 2832. 3234.  
## # i 355 more rows
```