

dev

Uli Kaulfuß

18 September 2019

```
## Parsed with column specification:
## cols(
##   City = col_character(),
##   State = col_character()
## )

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   location = col_character()
## )

## See spec(...) for full column specifications.

## Parsed with column specification:
## cols(
##   Year = col_double(),
##   `Geographic Region, Based on state of death` = col_character(),
##   `State of Death` = col_character(),
##   `Age Group` = col_character(),
##   `Veteran Suicides` = col_character(),
##   `Veteran Suicide Rate per 100,000` = col_character(),
##   `General Population Suicides` = col_character(),
##   `General Population Rate per 100,000` = col_character()
## )

## Parsed with column specification:
## cols(
##   State = col_character(),
##   Abbreviation = col_character()
## )
```

Development of CapVet App

```
df_cities <- as.data.frame(unique(df_sail_data$location))
names(df_cities) = c("City")

res <- left_join(df_cities, df_lookup_city, by='City')
```

```
## Warning: Column `City` joining factor and character vector, coercing into
## character vector
```

```

# clean for individual states only
df_suicide_data_rel <- df_suicide_data[df_suicide_data$State != 'Total U.S.' & df_suicide_data$State != 'Total']

# join state abbreviations
df_suicide_base <- left_join(df_suicide_data_rel, df_states_mapping, by='State')
df_suicide_base <- df_suicide_base %>% select(-State) %>% rename(State = "Abbreviation")

#rename columns

df_suicide_base <- rename(df_suicide_base, VetSuicideRatePer100k=`Veteran Suicide Rate per 100,000`)
df_suicide_base <- rename(df_suicide_base, GenPopSuicideRatePer100k=`General Population Rate per 100,000`)

clean_rate <- function(x, col){

  val <- x[col]

  val <- str_replace_all(val, "[*]", "")
  val <- str_replace_all(val, "-", "")

  val <- as.numeric(val)

}

# clean suicide rates
df_suicide_base$VetSuicideRatePer100k <- apply(df_suicide_base, 1, function(x) clean_rate(x, col="VetSuicideRatePer100k"))
df_suicide_base[is.na(df_suicide_base$VetSuicideRatePer100k),]$VetSuicideRatePer100k <- 0

df_suicide_base$GenPopSuicideRatePer100k <- apply(df_suicide_base, 1, function(x) clean_rate(x, col="GenPopSuicideRatePer100k"))
df_suicide_base[is.na(df_suicide_base$GenPopSuicideRatePer100k),]$GenPopSuicideRatePer100k = 0

# focus on state-based rates overall only
df_suicide_base <- df_suicide_base[df_suicide_base$`Age Group` == 'Total', ]

# subset on the ones where suicide rate > 0
df_suicide_base <- df_suicide_base[df_suicide_base$VetSuicideRatePer100k > 0, ]

# focus on 2016

df_sail_data_sub <- df_sail_data[df_sail_data$year == 2016,]

# lookup state information to location
df_sail_data_sub <- left_join(df_sail_data_sub, df_lookup_city, by=c('location' = 'City'))

# aggregate metrics for state-level
df_sail_agg <- df_sail_data_sub %>%
  filter(!is.na(State)) %>%
  group_by(State) %>%
  summarize(
    SMR30 = mean(`Acute care 30-day Standardized Mortality Ratio (SMR30)`, na.rm=TRUE),
    CallCenterAbandonmentRate = mean(`Call center abandonment rate`, na.rm=TRUE),
    GetUrgentAppointment_PCMH = mean(`Get an urgent care appointment as soon as needed (PCMH)`, na.rm=TRUE),
    NewMentalAppointments30DaysFromPreferredDate = mean(`New mental health appointments completed within 30 days of preferred date`, na.rm=TRUE)
  )

```

```

df_sail_agg_normal_metrics <- df_sail_agg %>%
  select(-State) %>%
  mutate_all(scale)

df_sail_agg_normal <- cbind(df_sail_agg %>% select(State), df_sail_agg_normal_metrics)

df_suicide_base_2016 <- df_suicide_base[df_suicide_base$Year == 2016,]

df_base <- full_join(df_suicide_base_2016, df_sail_agg_normal, by="State")

# account for missing values

df_base <- na.omit(df_base)

lm_model <- lm(VetSuicideRatePer100k ~ SMR30+CallCenterAbandonmentRate+GetUrgentAppointment_PCMH+NewMen
summary(lm_model)

```

```

##
## Call:
## lm(formula = VetSuicideRatePer100k ~ SMR30 + CallCenterAbandonmentRate +
##      GetUrgentAppointment_PCMH + NewMentalAppointments30DaysFromPreferredDate,
##      data = df_base)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.971  -5.315  -1.060   3.520  28.084
##
## Coefficients:
##              Estimate Std. Error t value
## (Intercept)      31.7667585   1.2730727   24.953
## SMR30             1.6106023   1.3380257    1.204
## CallCenterAbandonmentRate
##      -0.2434947   1.4608914   -0.167
## GetUrgentAppointment_PCMH
##       0.0002892   1.3504696    0.000
## NewMentalAppointments30DaysFromPreferredDate -3.9985008   1.3420351   -2.979
##
##              Pr(>|t|)
## (Intercept)    < 2e-16 ***
## SMR30           0.23544
## CallCenterAbandonmentRate
##       0.86843
## GetUrgentAppointment_PCMH
##       0.99983
## NewMentalAppointments30DaysFromPreferredDate  0.00478 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.667 on 42 degrees of freedom
## Multiple R-squared:  0.2003, Adjusted R-squared:  0.1242
## F-statistic:  2.63 on 4 and 42 DF,  p-value: 0.04763

```

```

df_va_exp <- df_va_exp %>% select(-StateLong)

df_base_overtime <- left_join(df_suicide_base, df_va_exp, by=c("State"="State", "Year"="Year"))

```

```
df_base_overtime <- df_base_overtime %>% rename(GeographicRegion = `Geographic Region, Based on state o

df_base_overtime$GeographicRegion <- as.factor(df_base_overtime$GeographicRegion)
df_base_overtime$State <- as.factor(df_base_overtime$State)
```

Evaluate SVM and RandomForest as regressions approximating veteran suicide rates.

```
df_model <- df_base_overtime %>% select(VetSuicideRatePer100k,GeographicRegion,State,MedicalAndGOE,Veter

df_model <- na.omit(df_model)

model_eval <- performanceEstimation(
  PredTask(VetSuicideRatePer100k ~ ., df_model),
  c(workflowVariants(learner="svm",
    learner.pars=list(cost=c(1,3,5),gamma=c(0.1,0.01))),
    workflowVariants(learner="randomForest",
    learner.pars=list(ntree=c(10,100,200)))
  ),
  EstimationTask(method=CV(nReps=3)))
```

```
##
##
## ##### PERFORMANCE ESTIMATION USING CROSS VALIDATION #####
##
## ** PREDICTIVE TASK :: df_model.VetSuicideRatePer100k
##
## ++ MODEL/WORKFLOW :: svm.v1
## Task for estimating all metrics of the selected evaluation function using
## 3 x 10 - Fold Cross Validation
## Run with seed = 1234
## Iteration :*****
##
##
## ++ MODEL/WORKFLOW :: svm.v2
## Task for estimating all metrics of the selected evaluation function using
## 3 x 10 - Fold Cross Validation
## Run with seed = 1234
## Iteration :*****
##
##
## ++ MODEL/WORKFLOW :: svm.v3
## Task for estimating all metrics of the selected evaluation function using
## 3 x 10 - Fold Cross Validation
## Run with seed = 1234
## Iteration :*****
##
##
## ++ MODEL/WORKFLOW :: svm.v4
## Task for estimating all metrics of the selected evaluation function using
## 3 x 10 - Fold Cross Validation
## Run with seed = 1234
## Iteration :*****
##
```

```
##
## ++ MODEL/WORKFLOW :: svm.v5
## Task for estimating all metrics of the selected evaluation function using
## 3 x 10 - Fold Cross Validation
## Run with seed = 1234
## Iteration :*****
##
##
## ++ MODEL/WORKFLOW :: svm.v6
## Task for estimating all metrics of the selected evaluation function using
## 3 x 10 - Fold Cross Validation
## Run with seed = 1234
## Iteration :*****
##
##
## ++ MODEL/WORKFLOW :: randomForest.v1
## Task for estimating all metrics of the selected evaluation function using
## 3 x 10 - Fold Cross Validation
## Run with seed = 1234
## Iteration :*****
##
##
## ++ MODEL/WORKFLOW :: randomForest.v2
## Task for estimating all metrics of the selected evaluation function using
## 3 x 10 - Fold Cross Validation
## Run with seed = 1234
## Iteration :*****
##
##
## ++ MODEL/WORKFLOW :: randomForest.v3
## Task for estimating all metrics of the selected evaluation function using
## 3 x 10 - Fold Cross Validation
## Run with seed = 1234
## Iteration :*****
```

```
summary(model_eval)
```

```
##
## == Summary of a Cross Validation Performance Estimation Experiment ==
##
## Task for estimating mae,mse,rmse,mape,nmse,nmae,theil using
## 3 x 10 - Fold Cross Validation
## Run with seed = 1234
##
## * Predictive Tasks :: df_model.VetSuicideRatePer100k
## * Workflows :: svm.v1, svm.v2, svm.v3, svm.v4, svm.v5, svm.v6, randomForest.v1, randomForest.v2, r
##
## -> Task: df_model.VetSuicideRatePer100k
## *Workflow: svm.v1
##
```

	mae	mse	rmse	mape	nmse	nmae
avg	3.3676339	23.597629	4.7981790	0.12182677	0.29891377	0.49569358
std	0.4232201	7.500073	0.7713222	0.01475438	0.06390031	0.04675627
med	3.3722866	22.635216	4.7574980	0.12101885	0.30341855	0.49159654
iqr	0.5434392	12.671800	1.3064462	0.01415105	0.07443936	0.04856087

```

## min      2.5939010 12.516652 3.5378881 0.08577607 0.18355486 0.36113524
## max      4.1637594 40.583450 6.3705141 0.15226866 0.46846666 0.63379518
## invalid  0.0000000 0.000000 0.0000000 0.00000000 0.00000000 0.00000000
##          theil
## avg      0.18503234
## std      0.04336257
## med      0.16690286
## iqr      0.06552804
## min      0.11838452
## max      0.26425343
## invalid  0.00000000
##
## *Workflow: svm.v2
##          mae      mse      rmse      mape      nmse      nmae
## avg      3.2451564 22.13326 4.6559083 0.11592977 0.28351741 0.47896744
## std      0.3535305  6.46691 0.6866521 0.01197543 0.06507950 0.04950829
## med      3.2060220 20.39952 4.5165544 0.11655838 0.28849125 0.47041612
## iqr      0.3392614 10.56636 1.1193090 0.01309501 0.07212415 0.05466689
## min      2.5089135 11.57988 3.4029224 0.08556357 0.17402591 0.35560049
## max      4.0492722 35.32533 5.9435119 0.13669878 0.45201312 0.61636827
## invalid  0.0000000 0.000000 0.0000000 0.00000000 0.00000000 0.00000000
##          theil
## avg      0.16531196
## std      0.04051846
## med      0.15191407
## iqr      0.06961922
## min      0.10658515
## max      0.23719958
## invalid  0.00000000
##
## *Workflow: svm.v3
##          mae      mse      rmse      mape      nmse      nmae
## avg      3.2257858 22.021530 4.6468413 0.11488372 0.28311616 0.47655246
## std      0.3253328  6.233226 0.6657086 0.01055706 0.06646506 0.04975701
## med      3.1729498 20.626409 4.5416296 0.11487490 0.28709826 0.46989977
## iqr      0.2798185  9.925582 1.0506665 0.01189342 0.08354709 0.06034095
## min      2.5053673 11.787441 3.4332842 0.08876923 0.17528391 0.36367497
## max      3.9658998 34.133323 5.8423731 0.13188806 0.44643526 0.60367757
## invalid  0.0000000 0.000000 0.0000000 0.00000000 0.00000000 0.00000000
##          theil
## avg      0.16191269
## std      0.03988979
## med      0.15017655
## iqr      0.06793301
## min      0.10228240
## max      0.23340251
## invalid  0.00000000
##
## *Workflow: svm.v4
##          mae      mse      rmse      mape      nmse      nmae
## avg      4.6242838 41.24065 6.333152 0.17602996 0.51315686 0.67710976
## std      0.6874597 14.34788 1.082063 0.02817480 0.07358694 0.04194733
## med      4.5525555 36.82024 6.067968 0.16944729 0.51436679 0.68516572
## iqr      0.9881988 17.07136 1.378524 0.02648117 0.09909761 0.07012553

```

```

## min      3.2675547 18.71732 4.326352 0.13035135 0.37228791 0.56300993
## max      6.4241794 82.56945 9.086773 0.23508452 0.65110369 0.74083607
## invalid  0.0000000 0.000000 0.0000000 0.00000000 0.00000000 0.00000000
##          theil
## avg      0.40937913
## std      0.08259711
## med      0.40661468
## iqr      0.10813330
## min      0.29452675
## max      0.63108007
## invalid  0.00000000
##
## *Workflow: svm.v5
##          mae      mse      rmse      mape      nmse      nmae
## avg      3.8915261 31.30690 5.5113180 0.14629555 0.39050617 0.57020548
## std      0.6129601 11.34031 0.9820517 0.02574913 0.07216567 0.04956362
## med      3.8014812 29.60278 5.4405533 0.14379961 0.39124952 0.57305626
## iqr      0.9859574 15.50037 1.4228770 0.02857739 0.06485559 0.05042018
## min      2.8567887 15.11949 3.8883790 0.10198592 0.23112744 0.42176153
## max      5.3144112 61.98992 7.8733677 0.20357493 0.52663077 0.66144218
## invalid  0.0000000 0.000000 0.0000000 0.00000000 0.00000000 0.00000000
##          theil
## avg      0.27129235
## std      0.06550637
## med      0.27505160
## iqr      0.06586137
## min      0.16347859
## max      0.44839234
## invalid  0.00000000
##
## *Workflow: svm.v6
##          mae      mse      rmse      mape      nmse      nmae
## avg      3.6167800 27.242735 5.1467495 0.13403063 0.34166197 0.53078365
## std      0.5418859 9.390701 0.8830029 0.02212717 0.06565592 0.04919811
## med      3.5430272 25.364378 5.0354159 0.13259106 0.34431704 0.52992082
## iqr      0.9670887 14.778845 1.4289171 0.02369064 0.05837095 0.03996200
## min      2.7686866 13.805748 3.7156086 0.08892226 0.20320241 0.37701851
## max      4.7297671 50.952975 7.1381353 0.18415474 0.48335563 0.64438874
## invalid  0.0000000 0.000000 0.0000000 0.00000000 0.00000000 0.00000000
##          theil
## avg      0.22511813
## std      0.05318980
## med      0.21569539
## iqr      0.06661694
## min      0.13744065
## max      0.35556966
## invalid  0.00000000
##
## *Workflow: randomForest.v1
##          mae      mse      rmse      mape      nmse      nmae
## avg      3.5421724 23.997415 4.8682478 0.13099231 0.31103352 0.52254775
## std      0.4103448 5.514884 0.5548323 0.01938401 0.06419483 0.05500866
## med      3.4945614 23.791256 4.8775803 0.12880971 0.30070004 0.51899628
## iqr      0.5890566 6.347467 0.6605730 0.02145337 0.10372815 0.06732083

```

```

## min      2.7681166 14.886984 3.8583655 0.09366548 0.21443954 0.41278534
## max      4.2599419 37.448005 6.1194775 0.18032127 0.44696574 0.63844330
## invalid  0.0000000 0.000000 0.0000000 0.00000000 0.00000000 0.00000000
##          theil
## avg      0.19359415
## std      0.04057972
## med      0.19159217
## iqr      0.06275544
## min      0.13382616
## max      0.27391161
## invalid  0.00000000
##
## *Workflow: randomForest.v2
##          mae      mse      rmse      mape      nmse      nmae
## avg      3.4166344 22.581136 4.7240753 0.12560113 0.29150867 0.50398357
## std      0.3629915  5.005959 0.5228390 0.01584304 0.05080558 0.04652679
## med      3.4070963 22.633096 4.7574209 0.12471582 0.28589743 0.50636787
## iqr      0.5771194  5.262588 0.5564535 0.01369492 0.05378751 0.05212059
## min      2.6788020 13.895228 3.7276303 0.08732181 0.19165018 0.36048061
## max      4.0381634 36.401184 6.0333393 0.16352995 0.39085935 0.58887084
## invalid  0.0000000 0.000000 0.0000000 0.00000000 0.00000000 0.00000000
##          theil
## avg      0.18506883
## std      0.03412408
## med      0.18128659
## iqr      0.06273285
## min      0.13031291
## max      0.24275369
## invalid  0.00000000
##
## *Workflow: randomForest.v3
##          mae      mse      rmse      mape      nmse      nmae
## avg      3.4129704 22.531748 4.7171395 0.12540278 0.29067858 0.50331774
## std      0.3723676  5.204845 0.5385259 0.01616790 0.05183673 0.04665485
## med      3.4058620 22.093193 4.7002984 0.12450949 0.28704097 0.50280901
## iqr      0.5727909  4.855912 0.5199411 0.01447930 0.05082299 0.05602924
## min      2.6730511 13.854348 3.7221429 0.08703258 0.18806378 0.35970673
## max      4.1242807 38.081733 6.1710399 0.16351107 0.39999265 0.59361940
## invalid  0.0000000 0.000000 0.0000000 0.00000000 0.00000000 0.00000000
##          theil
## avg      0.18453558
## std      0.0356834
## med      0.1756589
## iqr      0.0634722
## min      0.1270962
## max      0.2588805
## invalid  0.0000000
##
## Workflow Object:
## Workflow ID      :: svm.v4
## Workflow Function :: standardWF
## Parameter values:
## learner.pars  -> cost=1 gamma=0.01
## learner      -> svm

```


Train the SVM model with cost=1, gamma=0.01 as best performing parameters

```
svm_model <- svm(VetSuicideRatePer100k ~ ., data=df_model, cost=1, gamma=0.01)
```

Create linear model to approximate the veteran suicide rate per state over time depending on the expenditure of the VA

```
lm_model <- lm(VetSuicideRatePer100k ~ ., data=df_model)
summary(lm_model)
```

```
##
## Call:
## lm(formula = VetSuicideRatePer100k ~ ., data = df_model)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-17.7374	-2.2450	0.1833	2.0568	20.4174

```
##
## Coefficients: (3 not defined because of singularities)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.448e+01	3.562e+00	6.872	1.74e-11 ***
GeographicRegionNortheast	1.295e+01	3.449e+00	3.755	0.000192 ***
GeographicRegionSouth	7.506e+00	2.601e+00	2.886	0.004061 **
GeographicRegionWest	5.950e+00	3.280e+00	1.814	0.070226 .
StateAL	-7.013e+00	2.480e+00	-2.827	0.004865 **
StateAR	-2.485e+00	1.904e+00	-1.305	0.192352
StateAZ	2.312e+00	4.061e+00	0.569	0.569360
StateCA	-2.164e+01	1.493e+01	-1.450	0.147666
StateCO	3.795e+00	3.125e+00	1.215	0.225072
StateCT	-2.116e+01	2.311e+00	-9.156	< 2e-16 ***
StateDC	1.253e+00	3.547e+00	0.353	0.724016
StateDE	-1.186e+01	2.013e+00	-5.890	6.75e-09 ***
StateFL	-1.562e+01	1.154e+01	-1.353	0.176624
StateGA	-1.356e+01	4.712e+00	-2.879	0.004148 **
StateHI	-1.487e+01	1.864e+00	-7.977	8.89e-15 ***
StateIA	1.025e+00	2.319e+00	0.442	0.658661
StateID	8.551e+00	1.845e+00	4.634	4.50e-06 ***
StateIL	-9.053e+00	3.252e+00	-2.784	0.005562 **
StateIN	-1.918e+00	2.038e+00	-0.941	0.347009
StateKS	2.653e+00	2.382e+00	1.114	0.265911
StateKY	-1.414e+00	2.179e+00	-0.649	0.516643
StateLA	-9.979e+00	2.083e+00	-4.790	2.15e-06 ***
StateMA	-2.382e+01	3.367e+00	-7.074	4.67e-12 ***
StateMD	-1.669e+01	2.711e+00	-6.156	1.45e-09 ***
StateME	-7.488e+00	1.940e+00	-3.860	0.000127 ***
StateMI	-3.526e+00	2.753e+00	-1.281	0.200874
StateMN	-3.243e+00	1.822e+00	-1.780	0.075661 .
StateMO	2.349e+00	1.968e+00	1.194	0.233184
StateMS	-6.238e+00	1.814e+00	-3.438	0.000630 ***
StateMT	1.602e+01	1.804e+00	8.879	< 2e-16 ***
StateNC	-1.429e+01	4.813e+00	-2.968	0.003125 **
StateND	5.471e+00	3.421e+00	1.599	0.110376
StateNE	-1.339e+00	2.813e+00	-0.476	0.634196

```

## StateNH          -1.095e+01  1.902e+00  -5.756 1.44e-08 ***
## StateNJ          -2.292e+01  3.610e+00  -6.347 4.62e-10 ***
## StateNM           8.835e+00  2.048e+00   4.313 1.91e-05 ***
## StateNV           1.047e+01  2.244e+00   4.665 3.89e-06 ***
## StateNY          -2.790e+01  7.596e+00  -3.674 0.000263 ***
## StateOH          -6.670e+00  4.227e+00  -1.578 0.115139
## StateOK          -2.186e+00  2.112e+00  -1.035 0.301267
## StateOR           3.313e+00  2.753e+00   1.204 0.229280
## StatePA          -1.824e+01  7.332e+00  -2.487 0.013169 *
## StateRI          -1.726e+01  1.837e+00  -9.394 < 2e-16 ***
## StateSC          -9.388e+00  2.488e+00  -3.774 0.000179 ***
## StateSD           3.482e+00  3.236e+00   1.076 0.282387
## StateTN          -6.939e+00  3.145e+00  -2.206 0.027792 *
## StateTX          -2.051e+01  1.178e+01  -1.741 0.082261 .
## StateUT           6.483e+00  1.887e+00   3.435 0.000639 ***
## StateVA          -1.444e+01  4.763e+00  -3.032 0.002549 **
## StateVT           NA         NA         NA         NA
## StateWA          -7.492e+00  4.401e+00  -1.702 0.089285 .
## StateWI           NA         NA         NA         NA
## StateWV           NA         NA         NA         NA
## StateWY           1.982e+01  1.795e+00  11.045 < 2e-16 ***
## MedicalAndGOE     4.285e-06  7.896e-07   5.427 8.64e-08 ***
## VeteranPopulation  5.473e-06  8.224e-06   0.666 0.506013
## InsuranceAndIndemnities -6.470e-05  6.193e-05  -1.045 0.296623
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.378 on 544 degrees of freedom
## Multiple R-squared:  0.7787, Adjusted R-squared:  0.7572
## F-statistic: 36.12 on 53 and 544 DF, p-value: < 2.2e-16

```

[View\(df_model\)](#)