Looking for Trends

300958 Social Web Analysis

Week 11 Lab Solutions

- Complete the following R-code to write a function to compute Twitter's χ^2 statistic for trends.

```
TwitterChi = function(tweetsBefore, tweetsNow, N) {

  return((tweetsNow - tweetsBefore)^2/tweetsBefore
    + ((N - tweetsNow) - (N - tweetsBefore))^2/(N - tweetsBefore))

}
```

Using the matrix:


```
                 Before Now Sample Size
Number of Tweets     51
```


We can convert it int


```
> Y = matrix(0,2,2)
> Y[,1] = X[1:2]
> Y[,2] = X[3] - X[1:2]
> colnames(Y) = c("Topic","No Topic")
> rownames(Y) = c("Before","After")
```

We can stretch it into the long form:

```
> stretchTable = function(tab, variableNames) {
+     tabx = rep(rownames(tab), rowSums(tab))
+     l = ncol(tab)
+     m = nrow(tab)
+     cn = colnames(tab)
+     taby = c()
+     for (a in 1:m) {
+         for (b in 1:l) {
+             taby = c(taby, rep(cn[b], tab[a,b]))
+         }
+     }
```

```
+
+     d = data.frame(x = tabx, y = taby)
+     colnames(d) = variableNames
+     return(d)
+ }
>
> Z = stretchTable(Y, c("Time","Topic"))
```

And examine the top few items in the stretched list.

```
> head(Z)
```

```
    Time Topic
1 Before Topic
2 Before Topic
3 Before Topic
4 Before Topic
5 Before Topic
6 Before Topic
```


Then tabulate it to co


```
> table(Z)
```



```
        Topic
Time     No Topic Topic
  After       948    52
  Before      949    51
```

Compute the χ^2 randomisation distribution.

```
> ## define the functions to use
> expectedIndependent = function(X) {
+     n = sum(X)
+     p = rowSums(X)/sum(X)
+     q = colSums(X)/sum(X)
+     return(p %o% q * n) # outer product creates table
+ }
>
> chiSquaredStatistic = function(X, E) {
+     return(sum((X - E)^2/E))
+ }
>
```

```
> ## compute the expected table
> E = expectedIndependent(table(Z)) # compute expected counts if independent
>
> ## compute the randomisation distribution
> x2dist = replicate(1000, { # compute 1000 randomised chi-squared statistics
+   timeShuffle = sample(Z$Time)
+   topicShuffle = sample(Z$Topic)
+   Yindep = table(timeShuffle, topicShuffle)
+   chiSquaredStatistic(Yindep, E)
+ })
>
> hist(x2dist)
```


plot of chunk unnamed-chunk-7

Compute the χ^2 statistic for the original data.

```
> x2 = chiSquaredStatistic(table(Z), E)
```

Compute the p value of the test.

```
> ## pval is the proportion of x2dist that is greater than x2
> pval = mean(x2dist > x2)
> print(pval)
```

```
[1] 0.829
```

- Using R's χ^2 test:

```
> chisq.test(Y, simulate.p.value = TRUE)
```

```

    Pearson's Chi-squared test with simulated p-value (based on 2000
    replicates)

data:  Y
X-squared = 0.010236, df = NA, p-value = 1
```

- Linear regression:


```
> count = c(135, 145, 1
+ 149, 130, 107, 106, 8
+ 145, 120, 127, 110, 109, 122, 136, 125, 143, 177, 142
+ 173, 151, 174, 168, 158, 156, 128, 156, 140, 133, 132
> day = c(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)
>
> m = lm(sqrt(count) ~ day)
> summary(m)
```

```

Call:
lm(formula = sqrt(count) ~ day)

Residuals:
    Min      1Q  Median      3Q     Max
-2.6112 -0.5232  0.0051  0.5408  1.5870

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.80411    2.36352   1.186 0.241973
```

```
day          0.03714    0.01017   3.651 0.000703 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8862 on 43 degrees of freedom
Multiple R-squared:  0.2366,    Adjusted R-squared:  0.2189 
F-statistic: 13.33 on 1 and 43 DF,  p-value: 0.0007034
```

The above summary shows the intercept 2.80411, and the gradient 0.03714. The p-value for the gradient (0.000703), shows that the population gradient is not likely to be zero.

- Moving averages using a loop

The completed function is:

```
> windowWeights = function(m) {
+   if( m%%2 ) {
+      ## m is odd
+      w = rep(1,m)/m
+    } else {
+     ## m is even
+     w = c(0.5, rep(1,m-1), 0.5)/m
+   }
+ }
> 
> moving.average = function(x, m) {
+ 
+   ## compute window weights
+   w = windowWeights(m)
+ 
+   j = floor(m/2)
+   offsets = (-j):j
+   n = length(x)
+   res = rep(NA, n)
+   ## slide window and apply window weights to obtain averages
+   for(i in (j+1):(n-j)) {
+     res[i] = sum(w*x[i+offsets])
+   }
+   return(res)
+ }
```


- Make sure you understand this function and use it to compute the 7 point moving average of the tweet count data. Remember to square root first.

```
> trend = moving.average(sqrt(count), 7)
> print(trend)
```

```
 [1]       NA       NA       NA 11.03508 11.08952 11.11308 11.09440
 [8] 11.12934 11.13629 10.95317 10.88216 10.75223 10.46530 10.37510
[15] 10.50729 10.27367 10.58842 10.70917 10.84504 10.95310 11.02440
[22] 10.91278 11.16707 11.12873 11.12873 11.00570 11.14910 11.43977
[29] 11.64381 11.80603 11.99517 12.20817 12.36644 12.54253 12.49358
[36] 12.58693 12.71752 12.56672 12.47201 12.40686 12.16996 11.95962
[43]       NA       NA       NA
```

The following function computes a seasonal component by averaging over all observations that are m apart.

```
seasonal = function(x ,m) {
  ## extend the data so it is a multiple of m long
  tmp = c(x, rep(NA, m - length(x)%%m))
  ## convert to a matrix
  mat = matrix(tmp, nrow=m)
  ## Calculate the row means to get the seasonal component (excluding missing entries)
  seas = rowMeans(mat, na.rm=TRUE)
  seas = seas - mean(seas)
  return(seas)
}
```


- Make sure you understand this function an the seasonal component for the tweet count data. Reme e trend first.

```
> centred = sqrt(count) - trend # remove trend
> zcentred = centred[!is.na(centred)] # remove NAs at boundaries
> s = seasonal(zcentred, 7) # compute seasonal component
> print(s)
```

```
[1] -0.2961236 -0.7975847 -0.4794883  0.4469257  0.3593245  0.2767446
[7]  0.4902018
```

```
> plot(day, sqrt(count))
> lines(day, trend, col=2)
> lines(day[!is.na(centred)], trend[!is.na(centred)] + rep_len(s, length.out =
sum(!is.na(centred))))
```