Wind Revised

Peter Kinder

2023-03-19

Importing the data. Added month name.

```
Wind <- read.csv("full_monthly_storms.csv")
Wind$Month.1 <- as.factor(Wind$Month.1)
names(Wind) <- c("Month", "Windstorms", "Month.Name")
```

Decomposed time series in an effort to replicate Serena's approach.

```
decomposed <- decompose(ts(Wind$Windstorms, frequency = 12, start = c(2001, 1)))
Underlying.Wind <- data.frame(Month = seq(1, 240, 1), Underlying.Windstorms = na.omit(decomposed$trend))
```

Fit linear regression to decomposed trend in attempt to replicate Serena's approach. Interesting because it was similar, but not the same.

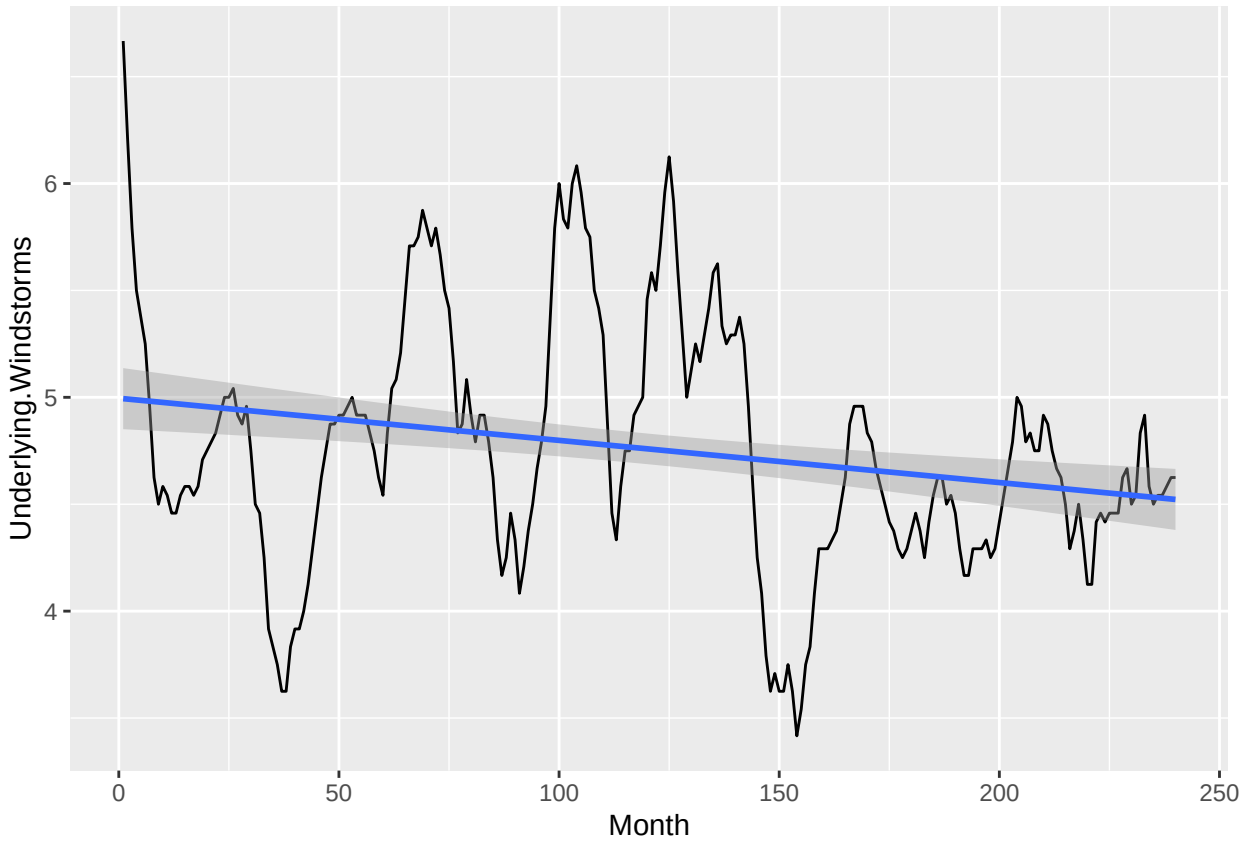
```
library(ggplot2)
Underlying.Wind.lm <- lm(Underlying.Windstorms ~ Month, data = Underlying.Wind)
summary(Underlying.Wind.lm)
```

```
##
## Call:
## lm(formula = Underlying.Windstorms ~ Month, data = Underlying.Wind)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29786 -0.35209 -0.02924  0.29417  1.67280
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.9958362  0.0729440  68.489  < 2e-16 ***
## Month       -0.0019724  0.0005248  -3.759 0.000215 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5633 on 238 degrees of freedom
## Multiple R-squared:  0.05603,    Adjusted R-squared:  0.05206
## F-statistic: 14.13 on 1 and 238 DF,  p-value: 0.0002151
```

```
ggplot(data = Underlying.Wind, aes(x = Month, y = Underlying.Windstorms)) + geom_line() + stat_smooth(m
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

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




```
y <- Wind$Windstorms

create.lags.df <- function(lags, y){
  df <- data.frame(y = y)
  for (i in 1:lags){
    index <- i + 1
    lag <- c(rep(NA, i), y[length(y):index])
    df <- cbind(df, lag)
  }
  names <- c("y")
  for (i in 1:lags){
    names <- c(names, paste("Lag", i, sep = ""))
  }
  names(df) <- names
  return(df)
}

months <- 36

df <- create.lags.df(months, y)
df <- na.omit(df)

lm.two.year <- glm(y ~ ., data = df, family = "poisson")
summary(lm.two.year)
```

```
##
## Call:
```

```
## glm(formula = y ~ ., family = "poisson", data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.59405  -0.63936   0.02259   0.46051   2.24883
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.6778228  0.4464029   3.759 0.000171 ***
## Lag1         0.0209985  0.0173372   1.211 0.225826
## Lag2         0.0414696  0.0176752   2.346 0.018966 *
## Lag3        -0.0017960  0.0183109  -0.098 0.921864
## Lag4        -0.0072270  0.0178949  -0.404 0.686315
## Lag5         0.0109144  0.0182286   0.599 0.549339
## Lag6        -0.0034023  0.0186918  -0.182 0.855566
## Lag7        -0.0103867  0.0181040  -0.574 0.566155
## Lag8        -0.0260564  0.0184572  -1.412 0.158034
## Lag9         0.0064748  0.0181436   0.357 0.721193
## Lag10        0.0192283  0.0185710   1.035 0.300484
## Lag11       -0.0024263  0.0182340  -0.133 0.894143
## Lag12       -0.0461760  0.0181456  -2.545 0.010936 *
## Lag13       -0.0095278  0.0177539  -0.537 0.591504
## Lag14        0.0018784  0.0177838   0.106 0.915878
## Lag15        0.0266014  0.0174888   1.521 0.128247
## Lag16       -0.0037841  0.0173530  -0.218 0.827378
## Lag17       -0.0052922  0.0172362  -0.307 0.758814
## Lag18        0.0043680  0.0171936   0.254 0.799459
## Lag19       -0.0298559  0.0173468  -1.721 0.085230 .
## Lag20        0.0092657  0.0177680   0.521 0.602032
## Lag21        0.0019249  0.0176037   0.109 0.912929
## Lag22       -0.0142762  0.0178371  -0.800 0.423499
## Lag23        0.0006318  0.0173063   0.037 0.970876
## Lag24       -0.0355525  0.0177409  -2.004 0.045071 *
## Lag25        0.0231173  0.0171518   1.348 0.177720
## Lag26        0.0170744  0.0173820   0.982 0.325952
## Lag27        0.0215245  0.0169186   1.272 0.203287
## Lag28        0.0271239  0.0172784   1.570 0.116460
## Lag29       -0.0095997  0.0175427  -0.547 0.584230
## Lag30       -0.0311358  0.0178166  -1.748 0.080538 .
## Lag31       -0.0436148  0.0179795  -2.426 0.015274 *
## Lag32       -0.0016375  0.0178888  -0.092 0.927065
## Lag33       -0.0033092  0.0180928  -0.183 0.854876
## Lag34       -0.0223013  0.0183559  -1.215 0.224388
## Lag35        0.0349804  0.0178527   1.959 0.050068 .
## Lag36        0.0015764  0.0181171   0.087 0.930663
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 247.59  on 215  degrees of freedom
## Residual deviance: 148.16  on 179  degrees of freedom
## AIC: 930.98
##
```

```
## Number of Fisher Scoring iterations: 4
```

```
preds <- predict(lm.two.year, type = "response")
sqrt(mean((preds - df$y)^2))
```

```
## [1] 1.707209
```

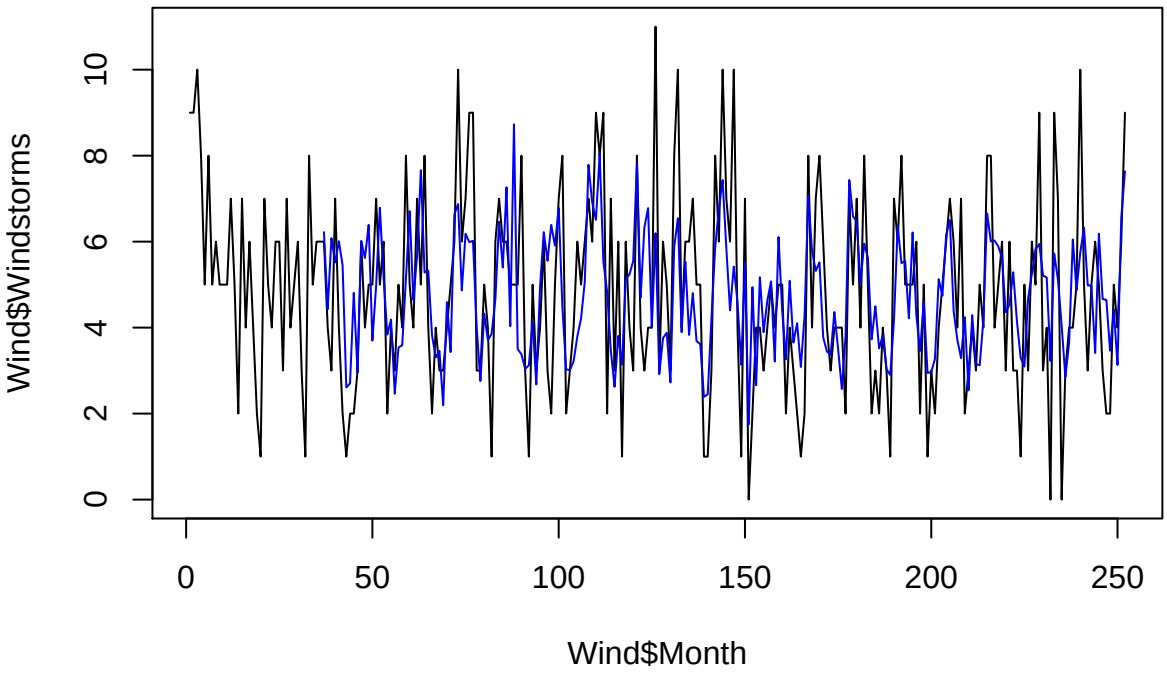
```
lm.aic <- step(lm.two.year, trace = 0, direction = c("both"))

summary(lm.aic)
```

```
##
## Call:
## glm(formula = y ~ Lag2 + Lag8 + Lag12 + Lag15 + Lag19 + Lag24 +
##     Lag25 + Lag30 + Lag31 + Lag35, family = "poisson", data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.84834  -0.63441  -0.00649   0.51899   2.13265
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.87812    0.23228   8.086 6.19e-16 ***
## Lag2         0.05259    0.01491   3.526 0.000422 ***
## Lag8        -0.03262    0.01557  -2.095 0.036187 *
## Lag12       -0.05023    0.01636  -3.070 0.002139 **
## Lag15        0.04054    0.01529   2.651 0.008019 **
## Lag19       -0.04288    0.01541  -2.783 0.005394 **
## Lag24       -0.03324    0.01550  -2.144 0.032011 *
## Lag25        0.02993    0.01499   1.997 0.045857 *
## Lag30       -0.02662    0.01510  -1.763 0.077897 .
## Lag31       -0.04208    0.01579  -2.665 0.007705 **
## Lag35        0.02587    0.01505   1.719 0.085601 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 247.59  on 215  degrees of freedom
## Residual deviance: 160.68  on 205  degrees of freedom
## AIC: 891.49
##
## Number of Fisher Scoring iterations: 4
```

```
preds <- predict(lm.aic, type = "response")
sqrt(mean((preds - df$y)^2))
```

```
## [1] 1.793199
```

```
plot(Wind$Month, Wind$Windstorms, type = "l")
lines(c(rep(NA, months), fitted(lm.aic)), col = "blue")
```


Wind$Windstorms
0
2
4
6
8
10
0
50
100
150
200
250
Wind$Month