Hw7

Stone Cai

2024-04-23

```r
library(macleish)
```

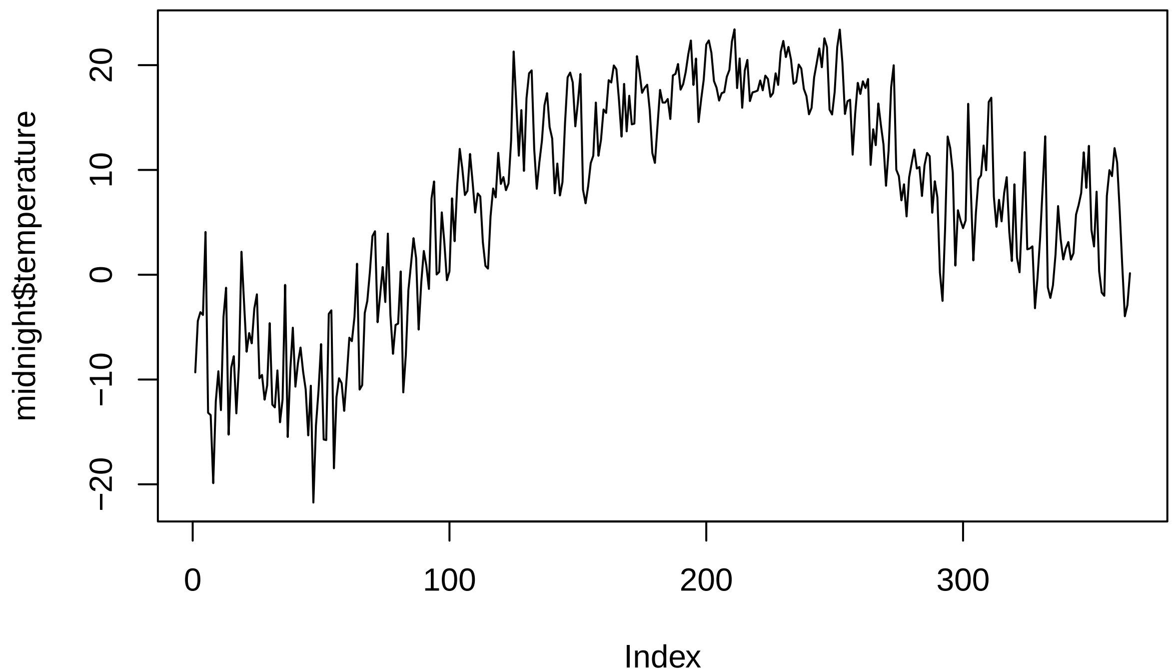
```
## Warning: package 'macleish' was built under R version 4.3.3

## Loading required package: etl

## Warning: package 'etl' was built under R version 4.3.3

## Loading required package: dplyr

## Warning: package 'dplyr' was built under R version 4.3.2

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3

## Warning: package 'ggplot2' was built under R version 4.3.2

## Warning: package 'tibble' was built under R version 4.3.2

## Warning: package 'tidyr' was built under R version 4.3.2

## Warning: package 'readr' was built under R version 4.3.2

## Warning: package 'purrr' was built under R version 4.3.2
```

```
## Warning: package 'stringr' was built under R version 4.3.2

## Warning: package 'forcats' was built under R version 4.3.2

## Warning: package 'lubridate' was built under R version 4.3.2

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.4.4     v stringr   1.5.0
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

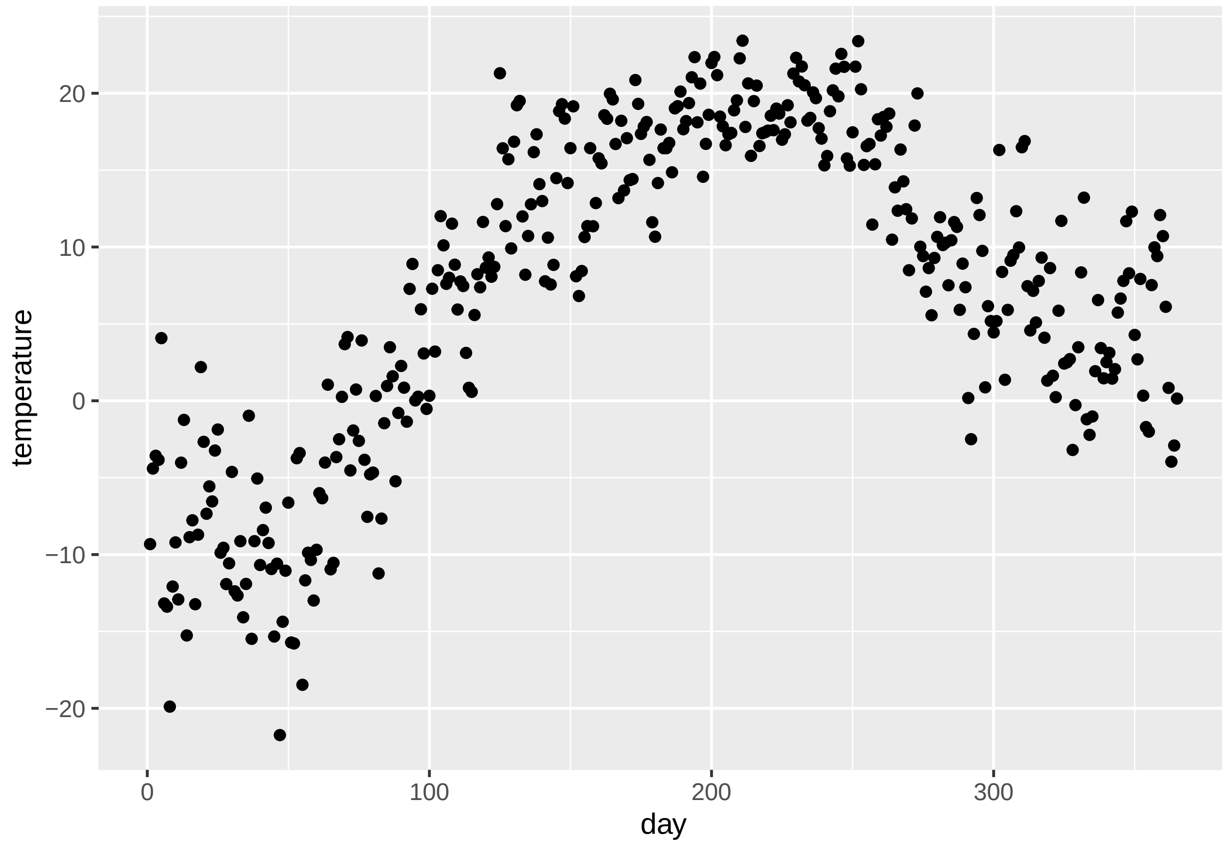
```r
library(dplyr)


midnight <- whately_2015 %>% dplyr::filter(substring(when, 12,19) == "")


plot(midnight$temperature, type = "l")
```



```
midnight<-midnight %>% mutate(day= 1:365)

View(midnight)
```

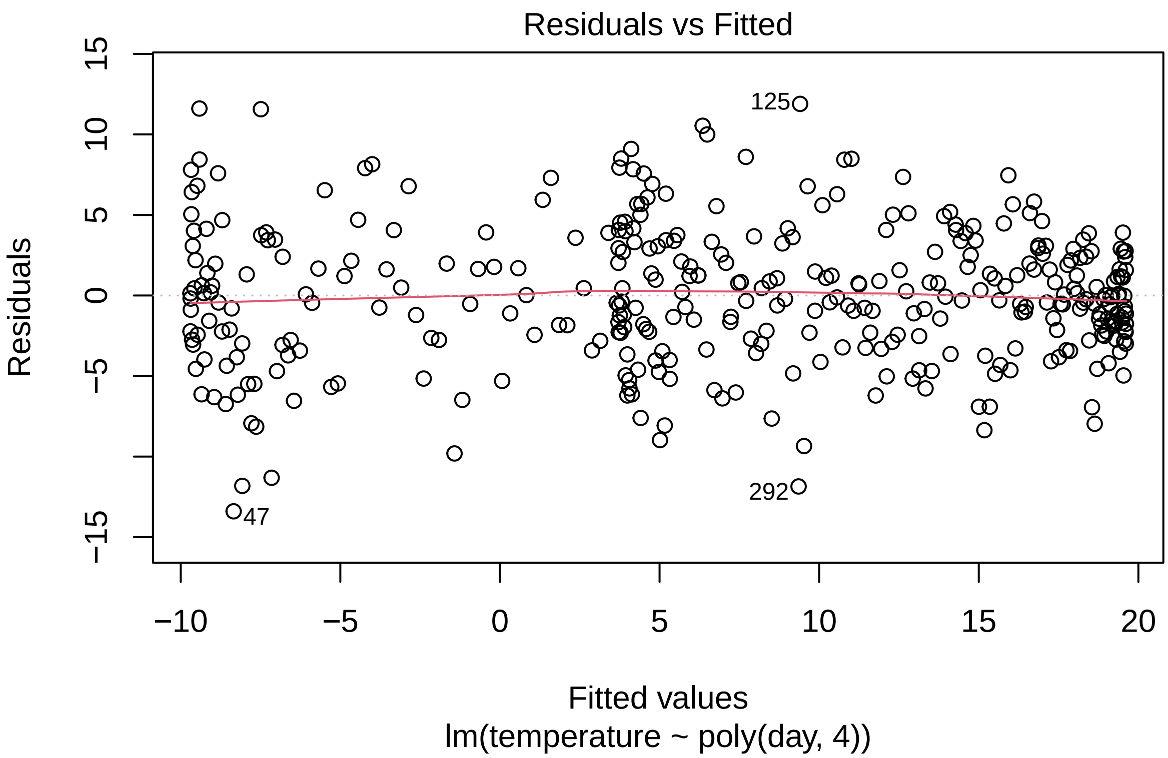
Here we plot midnight temperature for 365 days in a year.

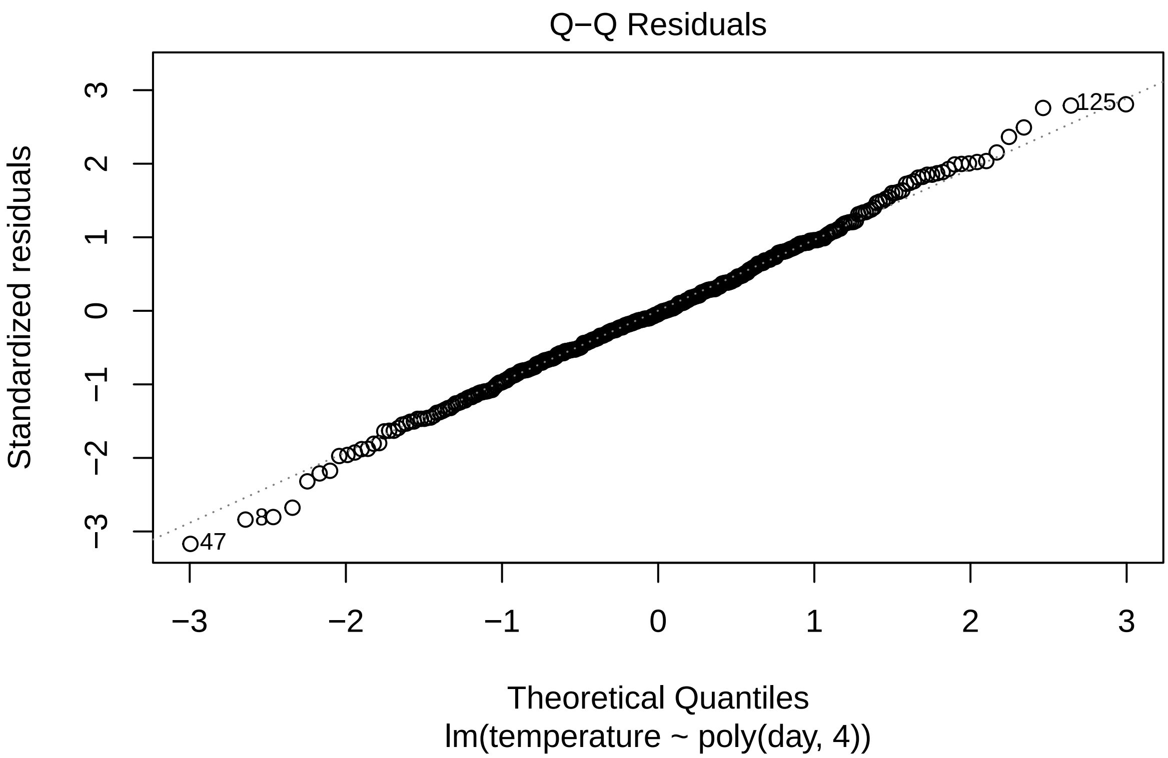
```
library(ggplot2)

ggplot(midnight,aes(x=day, y= temperature))+ geom_point()
```

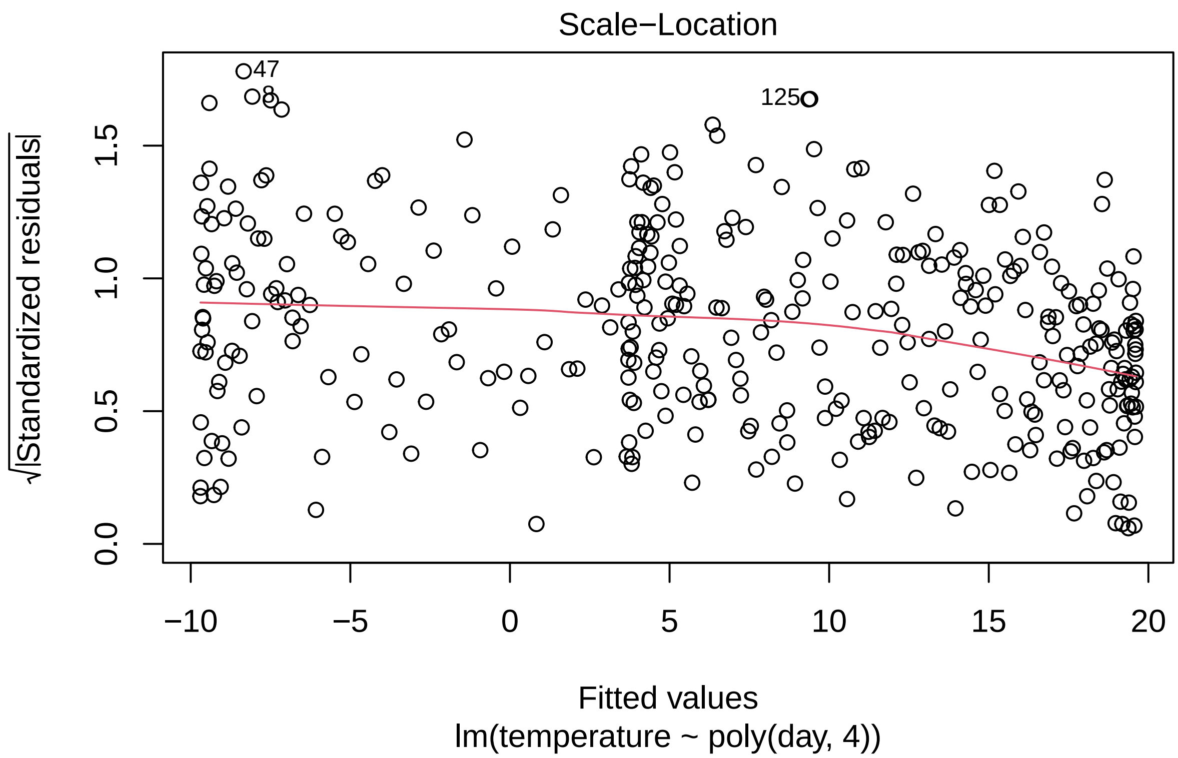


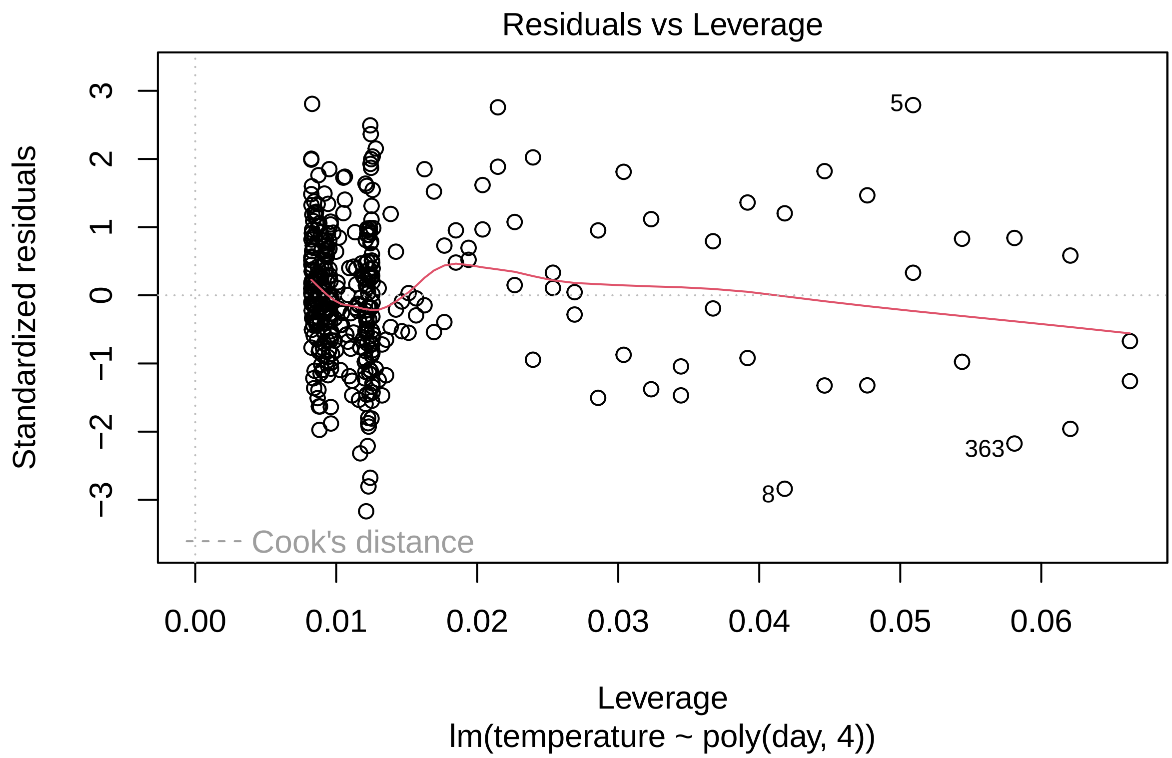
I plot temperature vs day just to see the shape of our data without time correlation.

```
# Fit linear model
poly_model <- lm(temperature ~ poly(day, 4), data = midnight)

plot(poly_model)
```

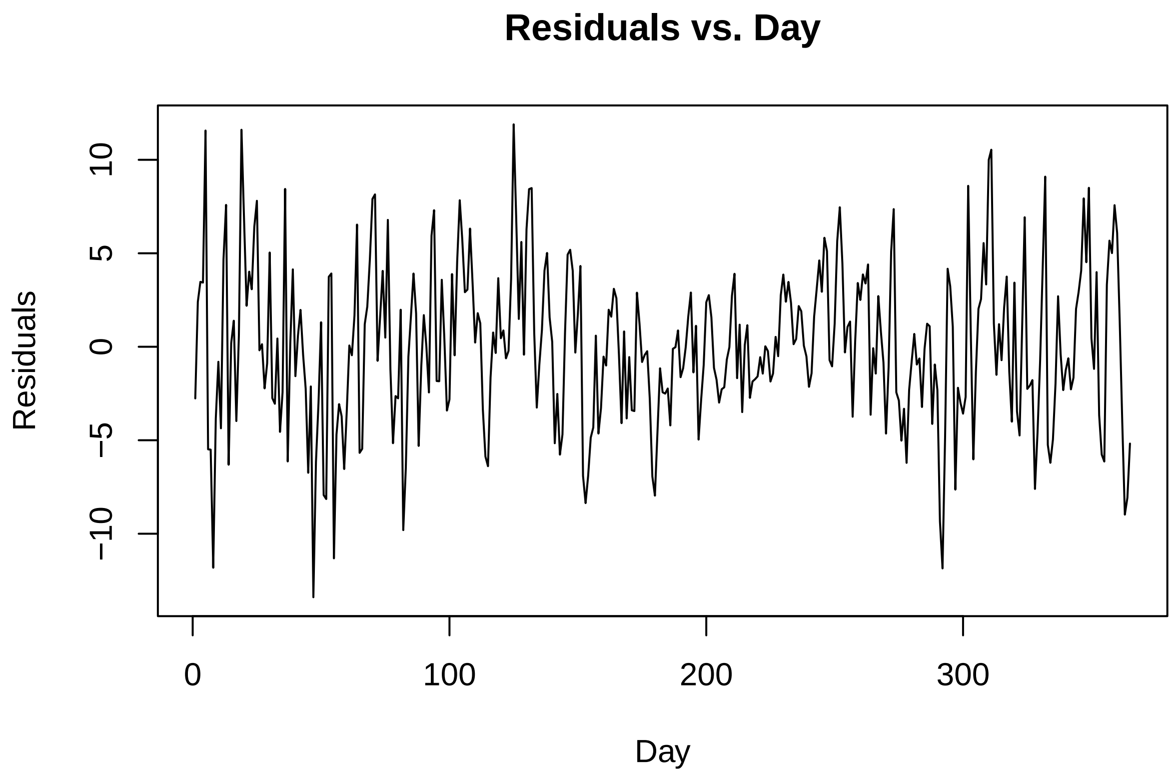

Residuals vs Fitted
Residuals
15
10
5
0
−5
−15
125
292
47
−10
−5
0
5
10
15
20
Fitted values
lm(temperature ~ poly(day, 4))


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(temperature ~ poly(day, 4))
−3
−2
−1
0
1
2
3
47
8
125


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−10
−5
0
5
10
15
20
47
8
125
Fitted values
lm(temperature ~ poly(day, 4))




```r
# Extract residuals
residuals <- resid(poly_model)

# Plot residuals vs. day
plot(midnight$day, residuals, type = "l", xlab = "Day", ylab = "Residuals", main = "Residuals vs. Day")
```

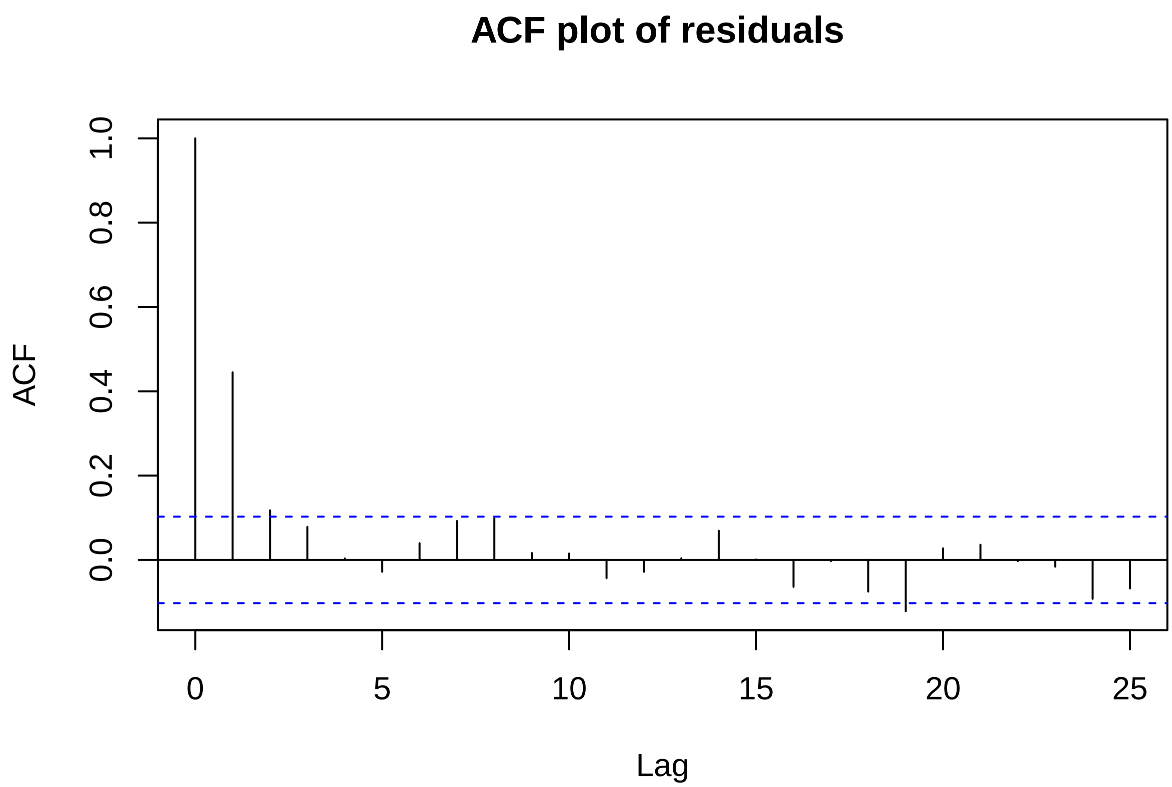


Constant Variance: In a stationary series, the variance of the residuals should remain relatively constant over time. The residuals seems to be relatively constant with no clear trends.

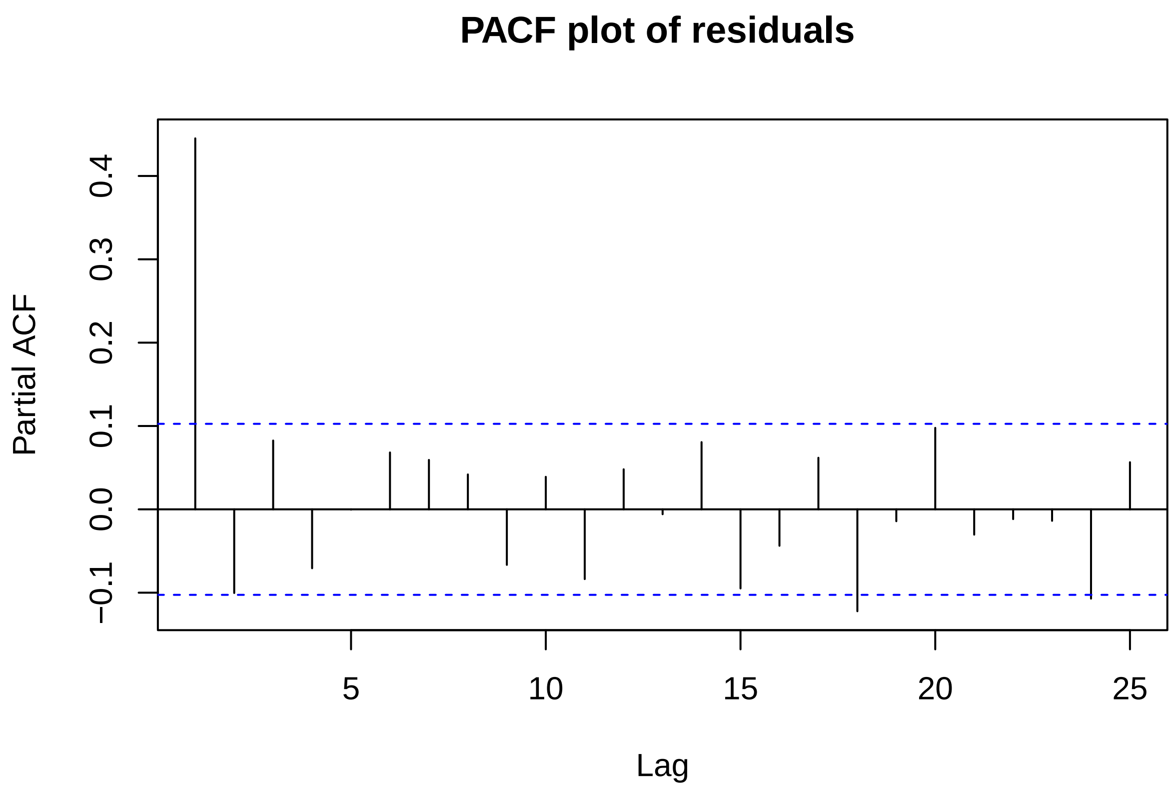
Random Fluctuations Around Zero: In a stationary series, the residuals should fluctuate randomly around zero. Any systematic pattern or trend in the residuals over time would indicate that the model is not stationary.

No Trend: A stationary series typically lacks any long-term trend. Here, the residuals do not show a consistent upward or downward trend over time.

```
acf(residuals, main="ACF plot of residuals ")
```




```r
pacf(residuals, main="PACF plot of residuals ")
```



The ACF plot quickly drops to zero which suggests it is stationary. ACF plots help determine the component Q of any Moving Average model. Here it looks like ACF drops off after lag 1 so Q = 1 from this graph. For the PACF plot, there is only one significant spike at lag 1. So P also = 1. This is the order of the AUTOREGRESSIVE portion of an Arima model. From part 1 in we didnt have to difference the residuals against day to get stationary model, so d=0. So Arima model should be (1,0,1)

```r
##install.packages("forecast")
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.3.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
# Fit ARIMA model to residuals
arima_model <- auto.arima(residuals)

summary(arima_model)
```

```
## Series: residuals
## ARIMA(1,0,1) with zero mean
##
## Coefficients:
##          ar1     ma1
```

```
##       0.1883  0.3198
## s.e.  0.1190  0.1179
##
## sigma^2 = 14.15:  log likelihood = -1000.63
## AIC=2007.25   AICc=2007.32   BIC=2018.95
##
## Training set error measures:
##                        ME     RMSE      MAE    MPE     MAPE     MASE
## Training set -0.003163974 3.751458 2.858562 89.304 184.3331 0.854457
##                      ACF1
## Training set -0.003539814
```

In summary, the ACF plot helps identify the order of the moving average (MA) component (q), while the PACF plot helps identify the order of the autoregressive (AR) component (p) in an ARIMA model. Running a formal ARIMA model function gives us the same results we achieved from our plots.