PS5

Hanyu Wang

The dataset 'problemset5.RData' (in the Problem Sets folder under Resources) contains data on positive laboratory tests for rotavirus ("rotaCA")* reported weekly to the National Respiratory and Enteric Viral Surveillance System (NREVSSLinks to an external site.) from the state of California for July 1991 to June 2006 ("time"). The rotavirus season in the northern hemisphere is typically considered to start in July and end in June; the vector "week" gives the week of the season (from 1 to 52) for the entire period of observation. Rotavirus vaccine was introduced in the US in 2006 with doses administered to infants at 2, 4, and 6 months of age.

*Note: This is not the "real" data from NREVSS (due to data sharing restrictions), but it is consistent with the actual data.

```
load("/Users/macbook/Desktop/problemset5.RData")
rota <- as.data.frame(problemset5)
```

```
length(rota$rotaCA) # 780 weeks
```

```
## [1] 780
```

```
length(rota$rotaCA)/52 # == 15 years
```

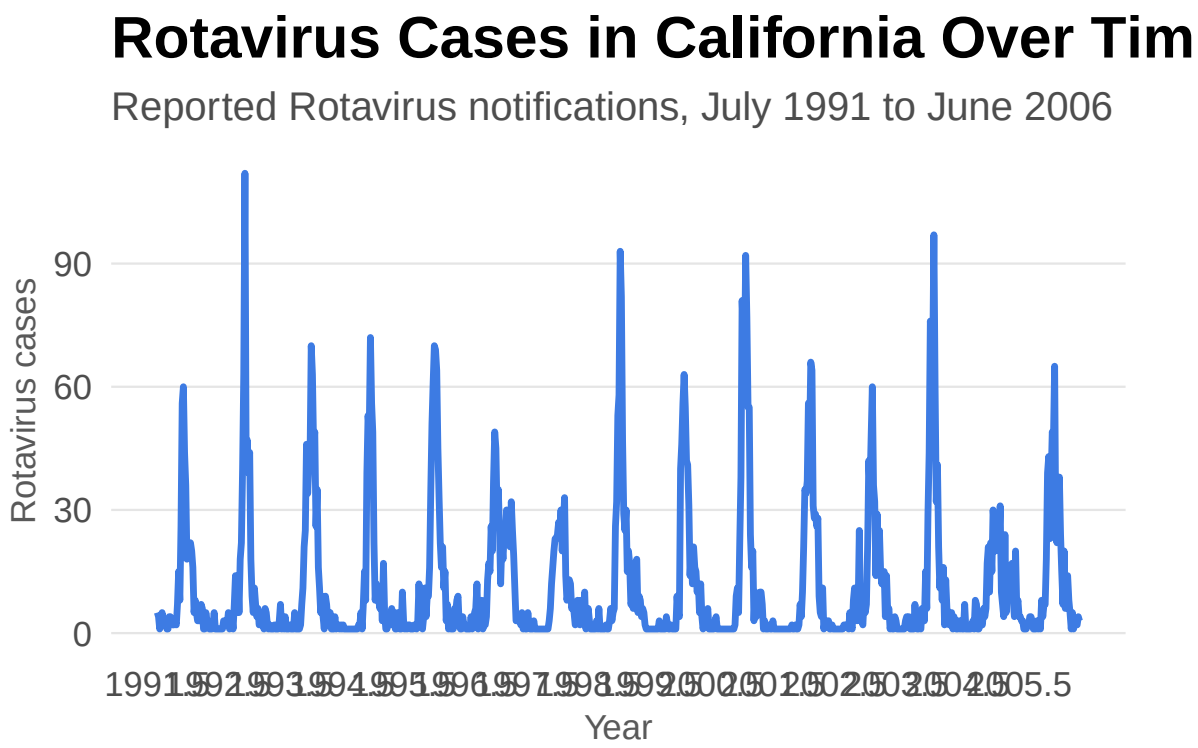
```
## [1] 15
```

1. What is (or are) the dominant period(s) of oscillation in the rotavirus time series? Is there any evidence that it changes over time? Show how you determined this.

i. Plot the number of rotavirus against "time".

```
rota %>%
  ggplot(aes(x = time, y = rotaCA)) +
  geom_line(linewidth = 1.2, color = "#3D7BE3") +
  scale_x_continuous(
    breaks = seq(min(rota$time), max(rota$time), by = 1)
  ) +
  theme_minimal(base_size = 16) +
  theme(
    plot.title = element_text(
      family = "Helvetica",
      face = "bold",
      size = 22,
      hjust = 0
    ),
    plot.subtitle = element_text(
      family = "Helvetica",
```

```
    size = 15,
    color = "gray30",
    hjust = 0
  ),
  plot.caption = element_text(
    family = "Helvetica",
    size = 10,
    color = "gray40",
    hjust = 1
  ),
  axis.text = element_text(
    family = "Helvetica",
    size = 13,
    color = "gray30"
  ),
  axis.title = element_text(
    family = "Helvetica",
    size = 13,
    color = "gray35"
  ),
  panel.grid.major.x = element_blank(),
  panel.grid.minor = element_blank(),
  panel.grid.major.y = element_line(color = "gray90", linewidth = 0.4),

  plot.margin = margin(25, 25, 25, 25)
) +
labs(
  title = "Rotavirus Cases in California Over Time",
  subtitle = "Reported Rotavirus notifications, July 1991 to June 2006",
  x = "Year",
  y = "Rotavirus cases"
)
```

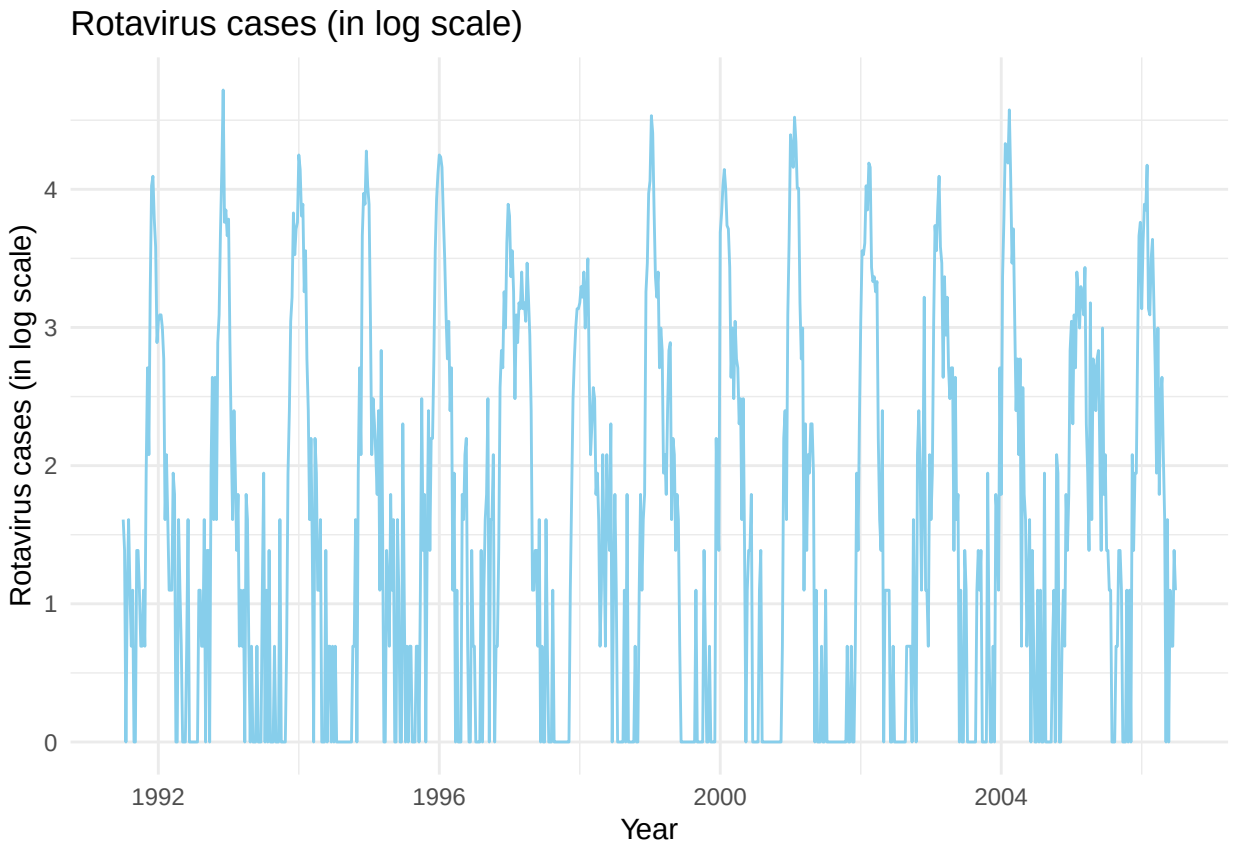


ii. Create a new variable "logrota" equal to the log of the number of bi-weekly rotavirus cases to make the time series more sinusoidal.

```
# log of the number of bi-weekly measles cases
rota$logrota <- log(rota$rotaCA)

# We check our data to see if we would run into this issue
range(rota$rotaCA) # Thankfully, we don't have zeroes
```

```
## [1]   1 112
```

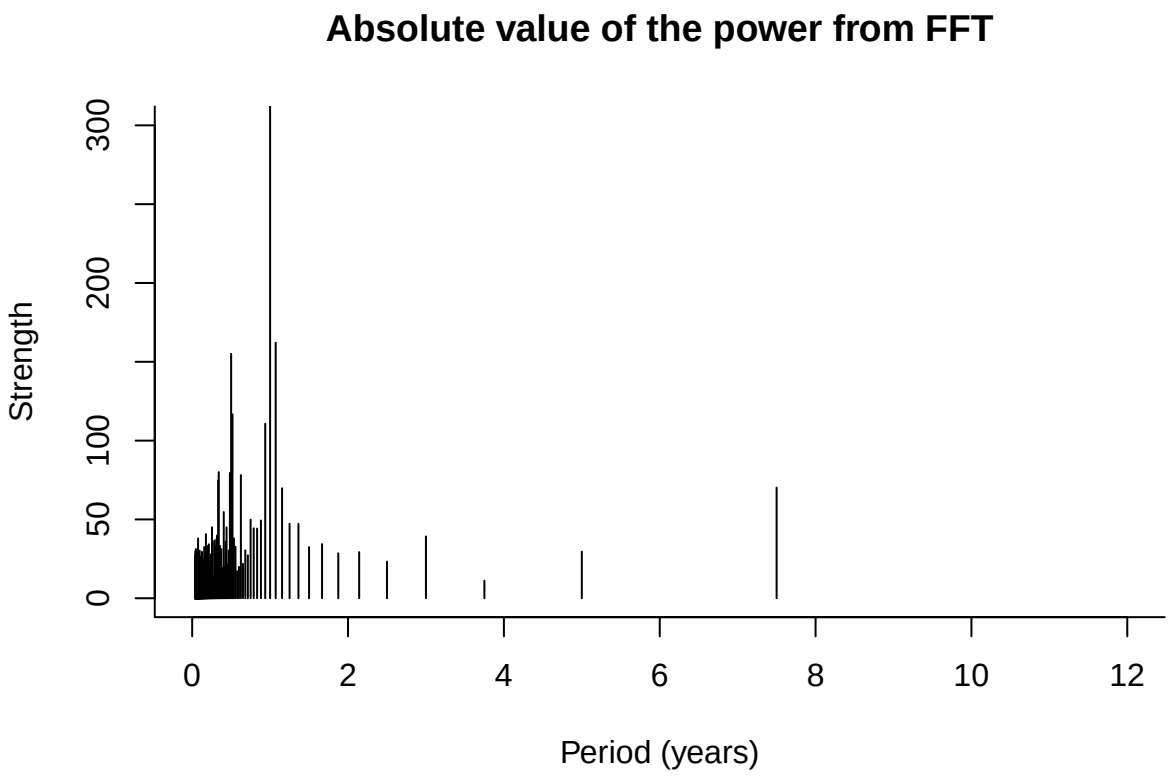
```
# log of the number of bi-weekly rotavirus cases
rota %>% ggplot(aes(x = time, y = logrota))+
  geom_line(color = "skyblue")+
  theme_minimal()+
  labs(y = "Rotavirus cases (in log scale)", x = "Year",
       title = "Rotavirus cases (in log scale)")
```



iii. Calculate and plot the absolute value of the power from the Fast Fourier Transform (FFT) for the log-transformed time series vs the period of oscillations (in # years).

Note: dt = 1/52 which reflects weekly notifications (i.e., 52 weekly notifications in a year)

```
# tmax: Number of biweeks
tmax <- length(rota$rotaCA)

# dt: time step in years
dt <- 1/52

# Calculate the absolute value of the power from FFT using a function fft()
ym <- abs(fft(rota$logrota))

# Frequency of oscillations (in weeks)
f <- (0:(tmax-1))/tmax

# Period of oscillations (in years) = 1 / frequency
t <- dt/f

# Make a plot
plot(x=t[2:round(tmax/2)+1],
     y=ym[2:round(tmax/2)+1],
     type='h', xlim=c(0,12),
     ylim=c(0,300), bty="l",
     xlab='Period (years)', ylab='Strength',
     main="Absolute value of the power from FFT")
```



Answer

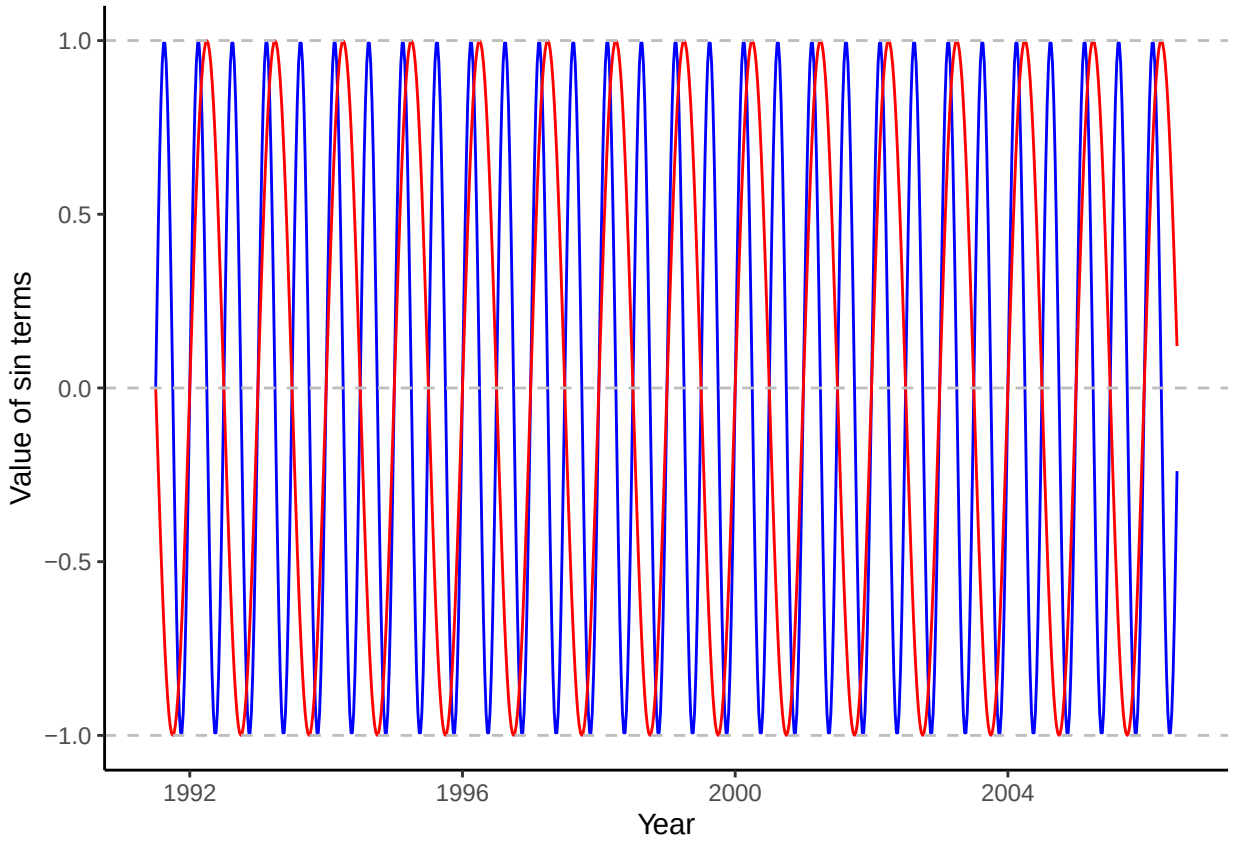
The dominant periods of oscillation in the rotavirus time series are 1 year and 6 months – the two most prominent peaks are one for a year and the other for half a year.

There is no evidence that it changes over time — the peak values remain stable around 1 year and 6 months with slight variations.

2. Fit a simple linear regression model with sine and cosine terms to account for the cycle(s) identified above to the log-transformed rotavirus time series.

```
# Create harmonic terms as follows:

# 6-month periods
cos6 <- cos( 4 * pi * rota$time)      # same as cos( 2* pi * time/0.5)
sin6 <- sin( 4 * pi * rota$time)      # same as sin( 2* pi * time/0.5)

# 12-month periods
cos12 <- cos( 2 * pi * rota$time ) # same as cos( 2* pi * time/1)
sin12 <- sin( 2 * pi * rota$time ) # same as sin( 2* pi * time/1)

# Plot harmonic terms (sin)
rota %>% ggplot(aes(x = time))+
  geom_line(aes(y = sin6),color = "blue")+
  geom_line(aes(y = sin12), color = "red")+
  geom_hline(yintercept=c(-1,0,1), linetype="dashed",
               color = "grey")+
```

```
  theme_classic()+
  labs(x='Year',y='Value of sin terms')
```




```
# Fit harmonic regression
mod6 <- glm(rota$logrota ~ cos6+sin6 + rota$time)
AIC(mod6)
```

```
## [1] 2666.566
```

```
mod12 <- glm(rota$logrota ~ cos12+sin12 + rota$time)
AIC(mod12)
```

```
## [1] 2042.184
```

```
mod <- glm(rota$logrota ~ cos6 + sin6 + cos12 + sin12 + rota$time)
AIC(mod)
```

```
## [1] 1963.905
```

```
summary(mod)
```

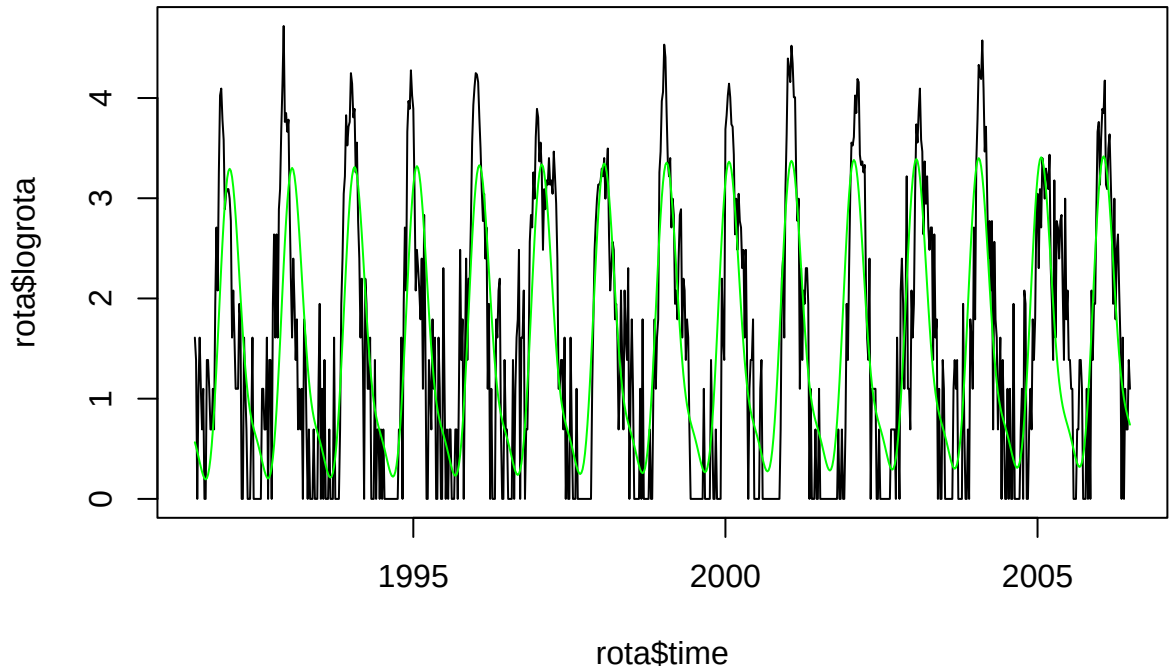
```
##
## Call:
## glm(formula = rota$logrota ~ cos6 + sin6 + cos12 + sin12 + rota$time)
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.146481  14.037235  -1.150    0.250
## cos6          0.357609   0.042929   8.330 3.66e-16 ***
## sin6          0.175596   0.042943   4.089 4.79e-05 ***
```

```
## cos12         1.263134   0.042929  29.424  < 2e-16 ***
## sin12         0.713562   0.042987  16.600  < 2e-16 ***
## rota$time     0.008846   0.007022   1.260    0.208
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.718725)
##
##     Null deviance: 1441.65  on 779  degrees of freedom
## Residual deviance:  556.29  on 774  degrees of freedom
## AIC: 1963.9
##
## Number of Fisher Scoring iterations: 2
```

```
# We can also calculate the predicted number of weekly measles cases as follows:
head(mod$fitted, 10)
```

```
##         1         2         3         4         5         6         7         8
## 0.5653388 0.5203391 0.4722573 0.4214076 0.3691440 0.3178210 0.2706562 0.2315448
##         9        10
## 0.2048087 0.1949276
```

```
plot(rota$time,rota$logrota,type='l')
lines(rota$time,mod$fitted,type='l',col='green')
```



(a) What is the amplitude and timing (phase shift/offset) of the seasonal cycle?

```
beta_sin6<-coef(mod)['sin6']
beta_cos6<-coef(mod)['cos6']
beta_sin12<-coef(mod)['sin12']
beta_cos12<-coef(mod)['cos12']
```

```
##### 6-month period #####

# Amplitude
amp6   <- sqrt(beta_sin6^2 +beta_cos6^2)
print(paste0("6-month period amplitude :", round(amp6,3)))
```

```
## [1] "6-month period amplitude :0.398"
```

```
# Phase angle
phase6 <- -atan(beta_sin6/beta_cos6)
print(paste0("6-month period phase angle :",round(phase6,3)))
```

```
## [1] "6-month period phase angle :-0.456"
```

```
##### 12-month period #####

# Amplitude
amp12   <- sqrt(beta_sin12^2 + beta_cos12^2)
print(paste0("1-year period amplitude :", round(amp12,3)))
```

```
## [1] "1-year period amplitude :1.451"
```

```
# Phase angle
phase12 <- -atan(beta_sin12/beta_cos12)
print(paste0("1-year period phase angle :", round(phase12,3)))
```

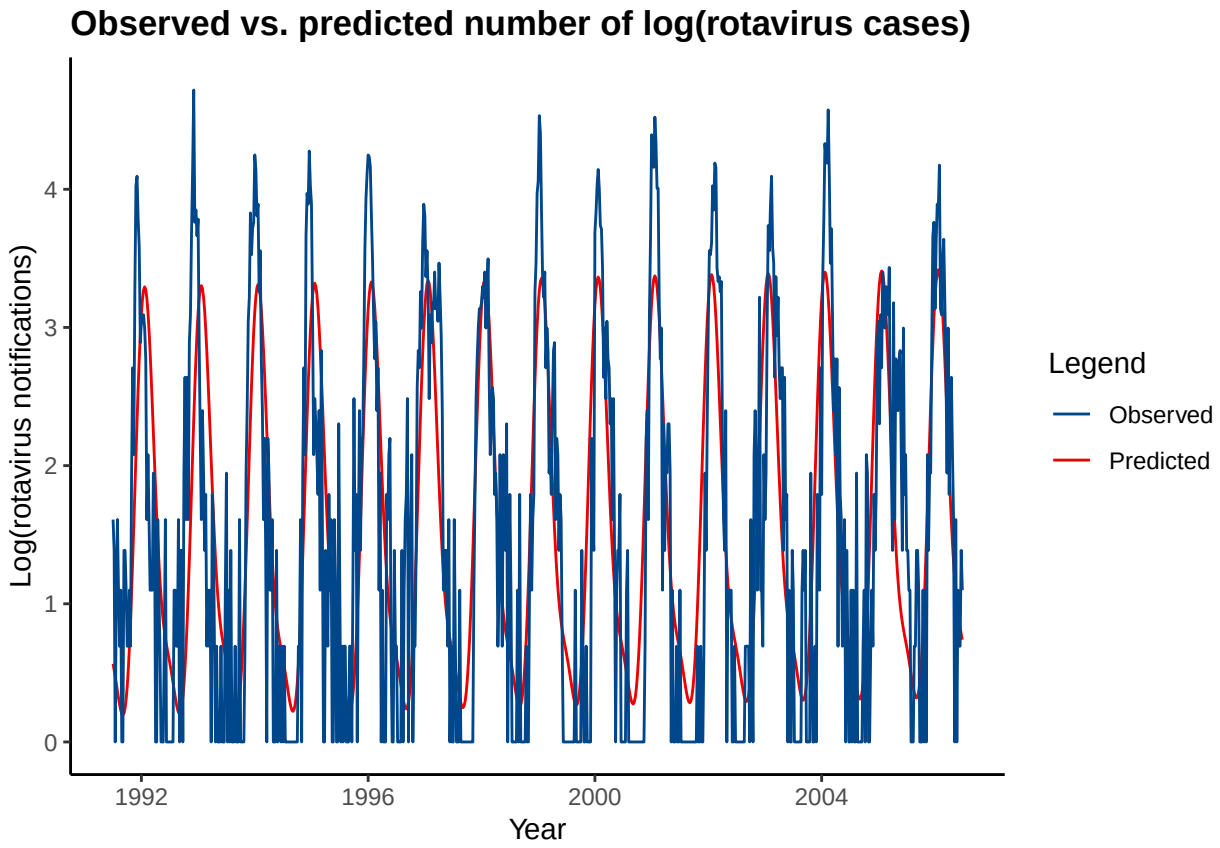
```
## [1] "1-year period phase angle :-0.514"
```

(b) Does the model provide an adequate fit to the time series? Why or why not?

```
# color setting
colors <- c("Observed" = "#00468BFF", "Predicted" = "#ED0000FF") # legend
```

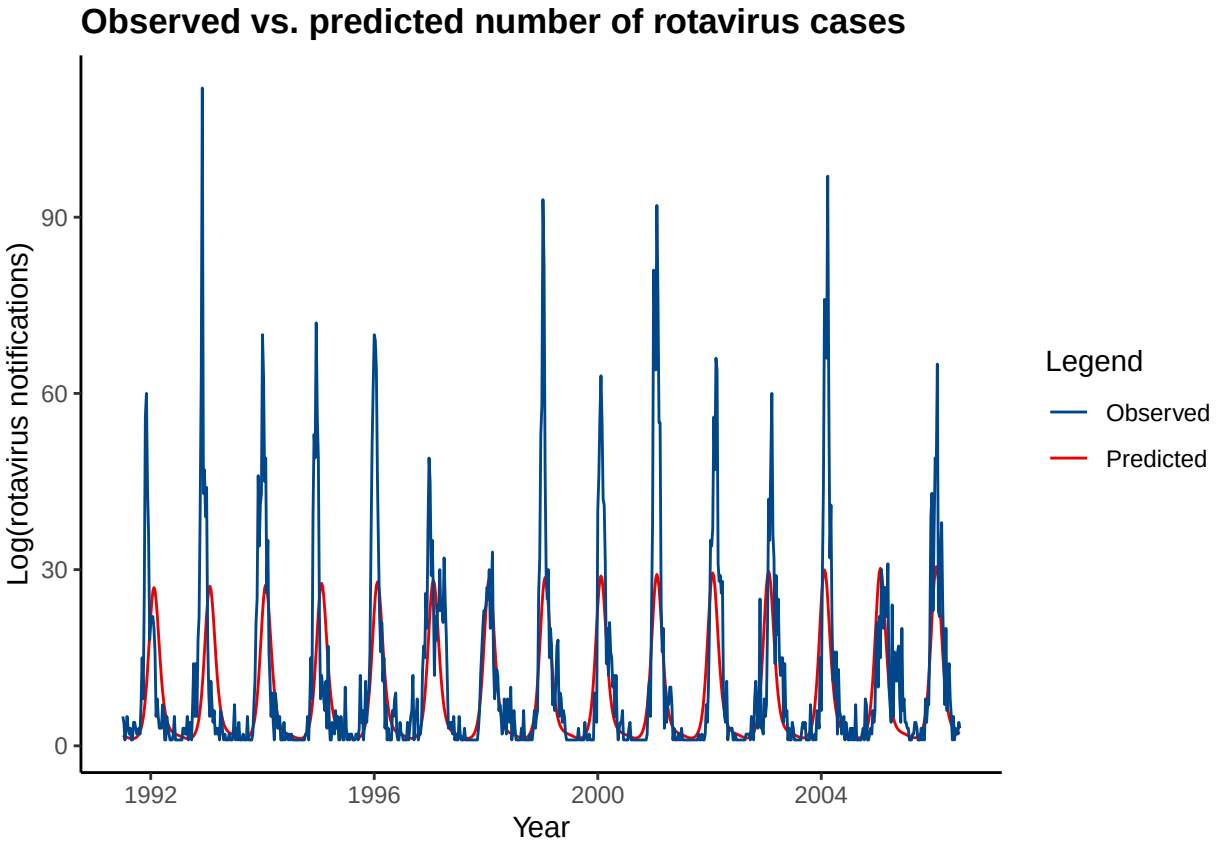
i. Plot this in the log scale

```
# ...and plot the prediction with the observed data.
# In log scale:
# Plot harmonic terms (sin)
rota %>% ggplot(aes(x = time))+
  geom_line(aes(y = mod$fitted, color = "Predicted"))+
  ylim(range(c(mod$fitted, rota$logrota)))+
  geom_line(aes(y = rota$logrota, color ="Observed"))+
  theme_classic()+
  labs(y = "Log(rotavirus notifications)", x = "Year",
       title = "Observed vs. predicted number of log(rotavirus cases)",
       color= "Legend")+
    scale_color_manual(values = colors)+
  theme(plot.title = element_text(face = "bold"))
```



ii. Plot this in original y-scale. We are exponentiating the fitted values since the harmonic regression used log scale values `logrota`.

```
## In original scale:
rota %>% ggplot(aes(x = time))+
  geom_line(aes(y = exp(mod$fitted), color = "Predicted"))+
  ylim(range(c(exp(mod$fitted), rota$rotaCA)))+
  geom_line(aes(y = rotaCA, color ="Observed"))+
  theme_classic()+
  labs(y = "Log(rotavirus notifications)", x = "Year",
       title = "Observed vs. predicted number of rotavirus cases",
       color= "Legend")+
  scale_color_manual(values = colors)+
  theme(plot.title = element_text(face = "bold"))
```

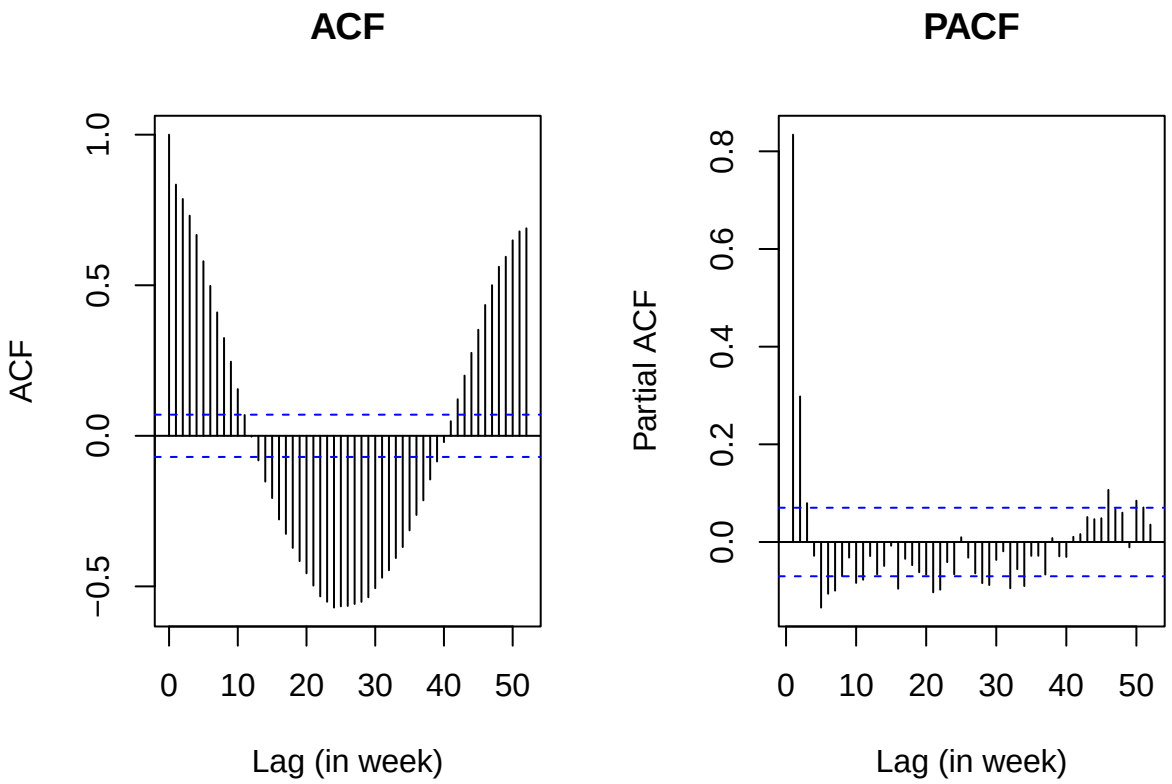


Answer:

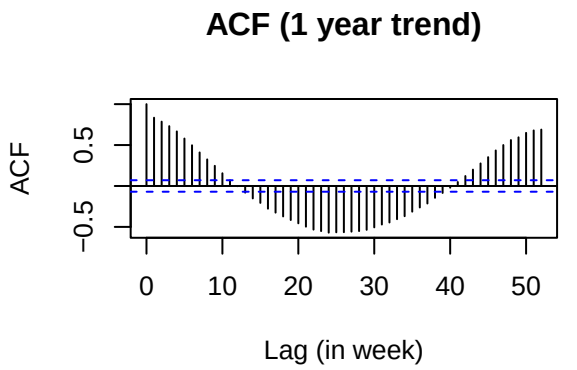
The model does not provide an adequate fit to the time series – although the periods and peak times of the red prediction line and the blue observation line are basically the same, the amplitude of the prediction curve is much smaller than that of the actual data.

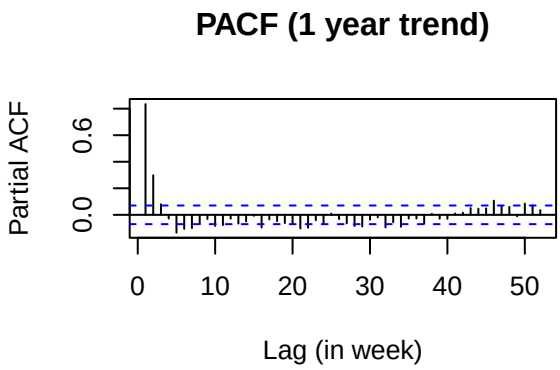
3. Calculate and plot the autocorrelation function (ACF) and partial autocorrelation function (PACF) for the log-transformed data. What terms (and associated lags) should you include in an ARIMA model? Explain your answer.

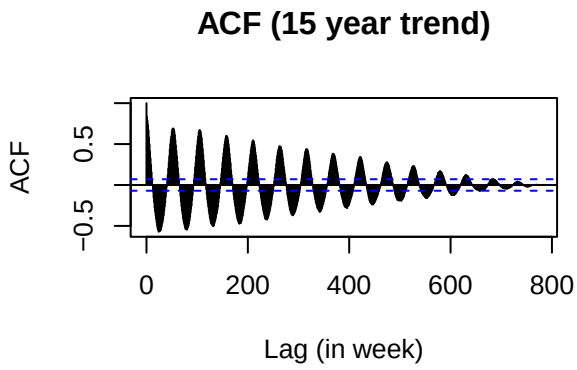
```
# Calculate and plot the autocorrelation (ACF) and
# the partial autocorrelation functions (PACF)
par(mfrow=c(1,2))
acf(rota$logrota, lag.max=52, main='ACF', xlab="Lag (in week)")
pacf(rota$logrota, lag.max=52, main='PACF', xlab="Lag (in week)")
```

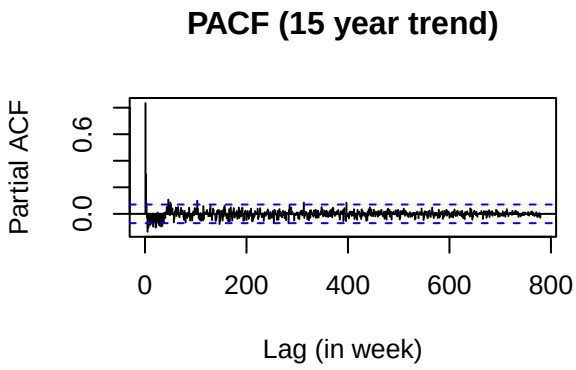



```
par(mfrow=c(2,2))
acf(rota$logrota, lag.max=52*1, main='ACF (1 year trend)',
    xlab="Lag (in week)")
pacf(rota$logrota, lag.max=52*1, main='PACF (1 year trend)',
     xlab="Lag (in week)")

acf(rota$logrota, lag.max=52*15, main='ACF (15 year trend)',
    xlab="Lag (in week)")
pacf(rota$logrota, lag.max=52*15, main='PACF (15 year trend)',
     xlab="Lag (in week)")
```

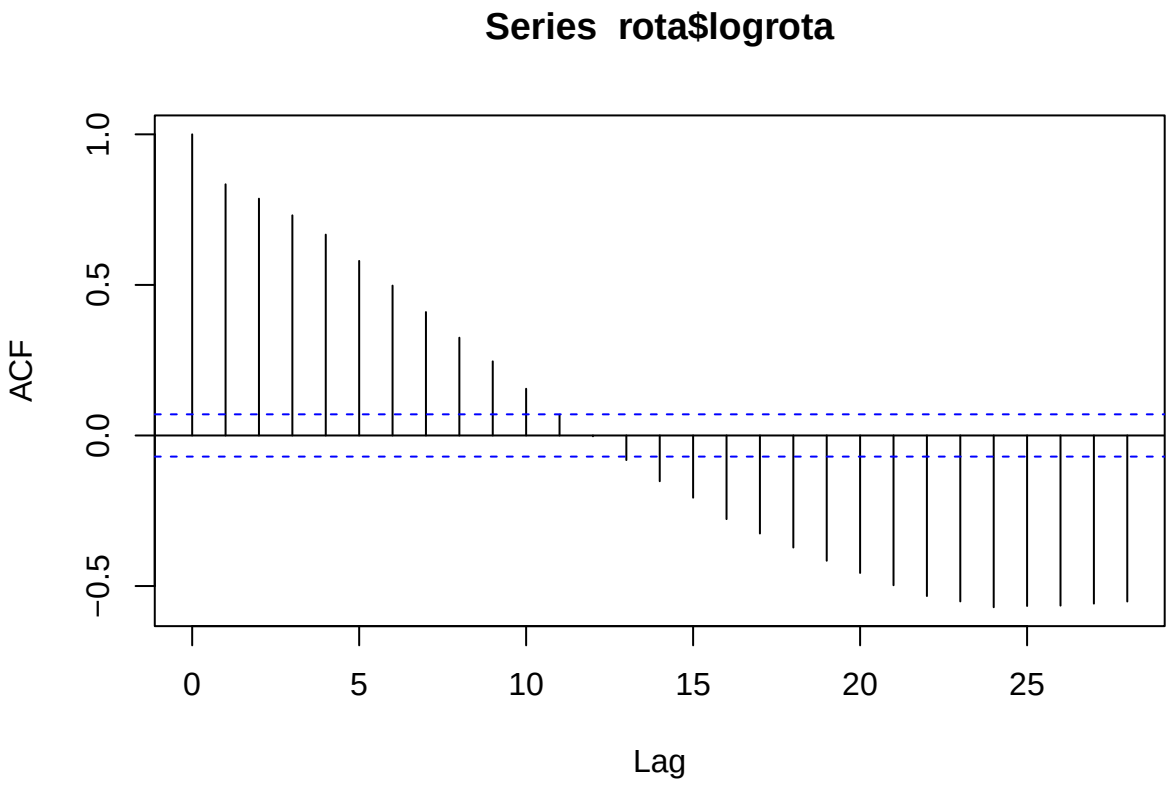




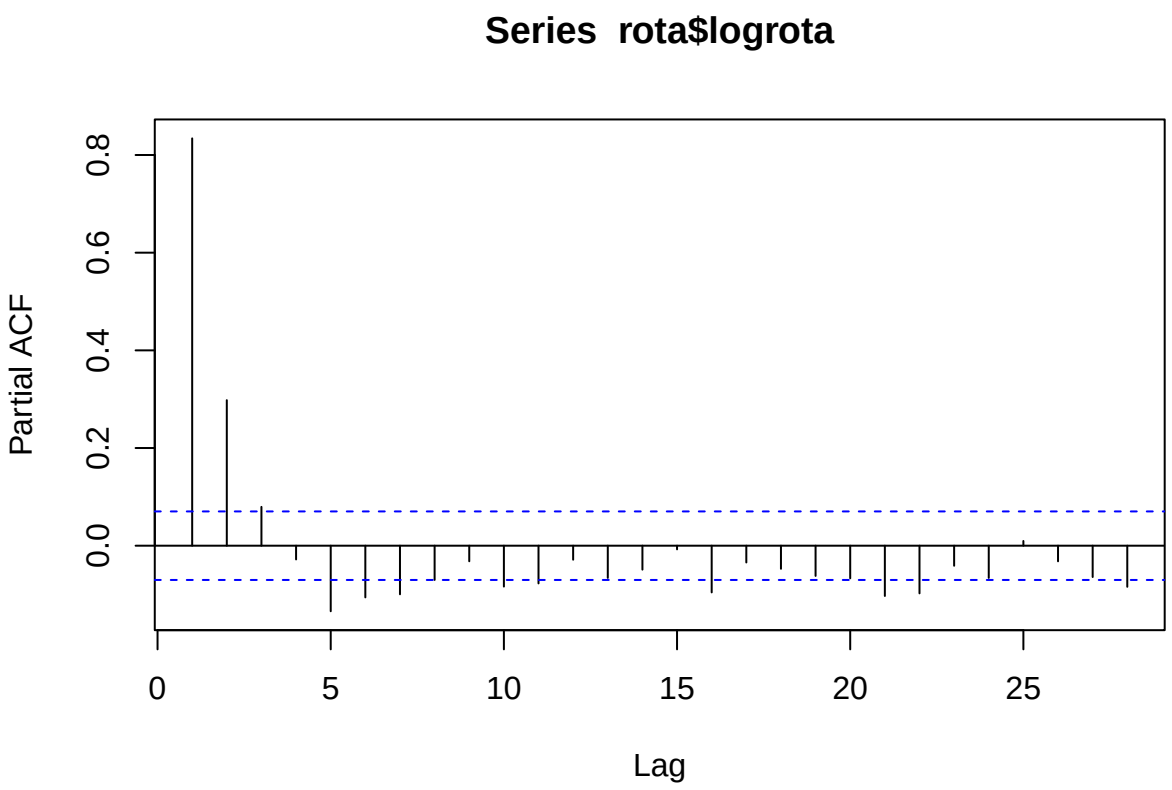





```r
# Optional: Various outputs are included in "acf"
acf <- acf(rota$logrota)
```



```r
pacf <- pacf(rota$logrota)
```




```r
names(acf)
```

```
## [1] "acf"    "type"   "n.used" "lag"    "series" "snames"
```

```r
names(pacf)
```

```
## [1] "acf"    "type"   "n.used" "lag"    "series" "snames"
```

Answer:

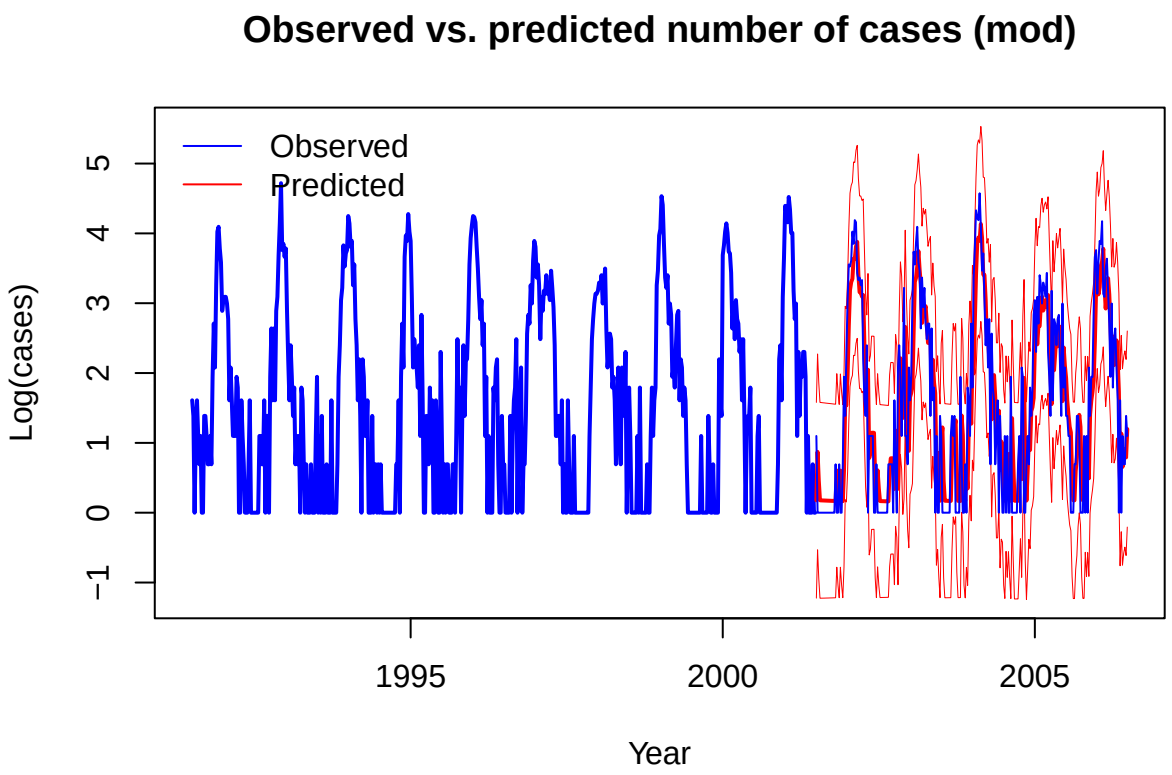
p AR terms should be included – ACF tails off gradually, while PACF cuts off after p lags.

The associated lags could be 2 (p=2) – PACF seems to cut off after 2 lags.

4. Fit the ARIMA model you describe in question 3 to the first 10 years of the log-transformed rotavirus time series and use it to predict the last 5 years of data. Does the model provide a good fit to the data? Explain your answer.

```r
# Using mod
# (This for loop may take long time to run.)
pred <- se <- c()
for (i in 1:(5*52)){
  # Fit the ARIMA model to the first 10 years of measles notifications
  mod <- arima(rota$logrota[1:(52*10+(i-1))],order=c(2,0,0))
  # NOTE: the default method ML will give you an error unfortunately.
  #       we cannot use it for our analysis.
```

```r
  # Predict the last 5 years of the data using the fitted model
  # (One-step forward prediction)
  predicted_mod <-predict(mod, n.ahead=1)
  pred[i] <- predicted_mod$pred # Point estimate
  se[i] <- predicted_mod$se # Standard error
  # print(i)
}

# Plot the observed vs. predicted number of cases
par(mfrow=c(1,1))
# Observed number of cases for the first 16 years
plot(x=rota$time[1:length(rota$time)],
     y=c(rota$logrota[1:(52*10)],rep(NA,5*52)),
     type='l',col='blue',
     ylab='Log(cases)',
     xlab='Year',main="Observed vs. predicted number of cases (mod)",
     lwd=2,ylim=range(c(pred-1.96*se,pred+1.96*se,rota$logrota)))

# Predicted number of cases (with 95% CI) for the last 5 years
lines(x=rota$time[(10*52+1):length(rota$time)], y=pred, col='red',lwd=2)
lines(x=rota$time[(10*52+1):length(rota$time)], y=pred+1.96*se, lwd=0.5, col='red')
lines(x=rota$time[(10*52+1):length(rota$time)], y=pred-1.96*se, lwd=0.5, col='red')

# Observed number of cases for the last 5 years
lines(x=rota$time[(10*52+1):length(rota$time)], y=rota$logrota[(52*10+1):(520+5*52)], col='blue')
legend(x="topleft",legend=c("Observed","Predicted"),col=c("blue","red"),lty=c(1,1),bty="n")
```



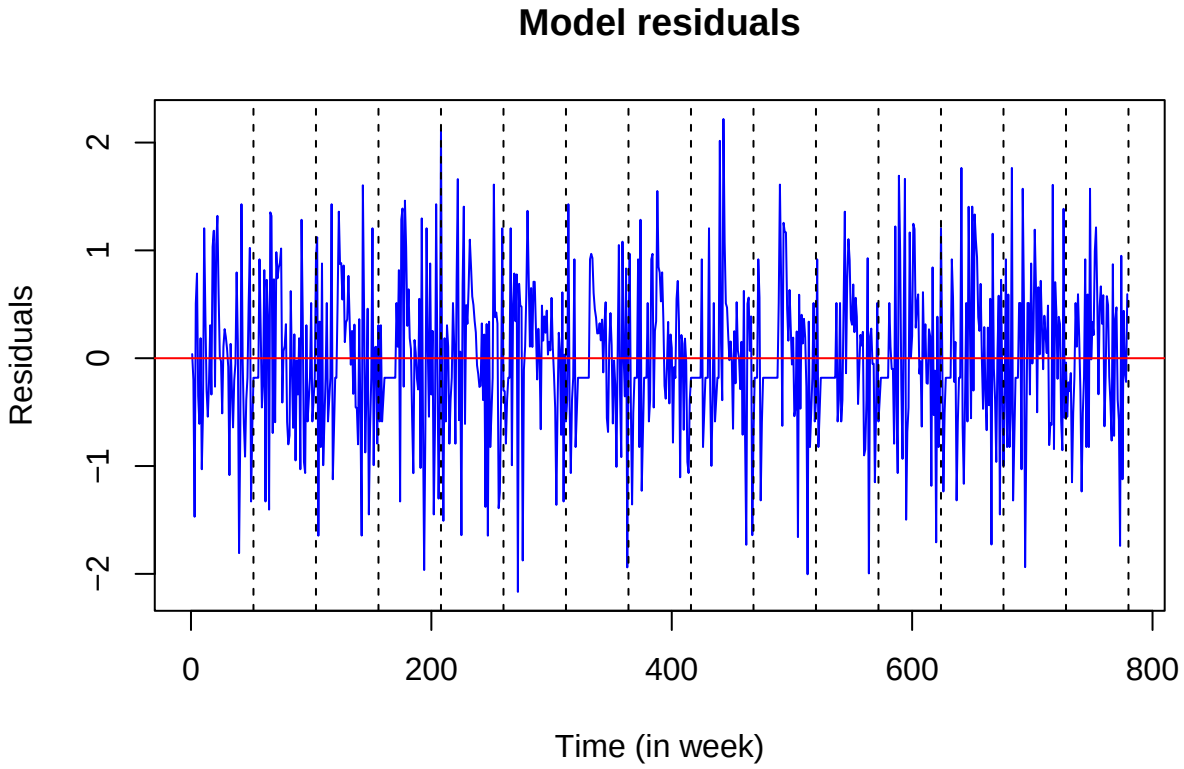
Answer:

Overall, the model provide a good fit to the data.

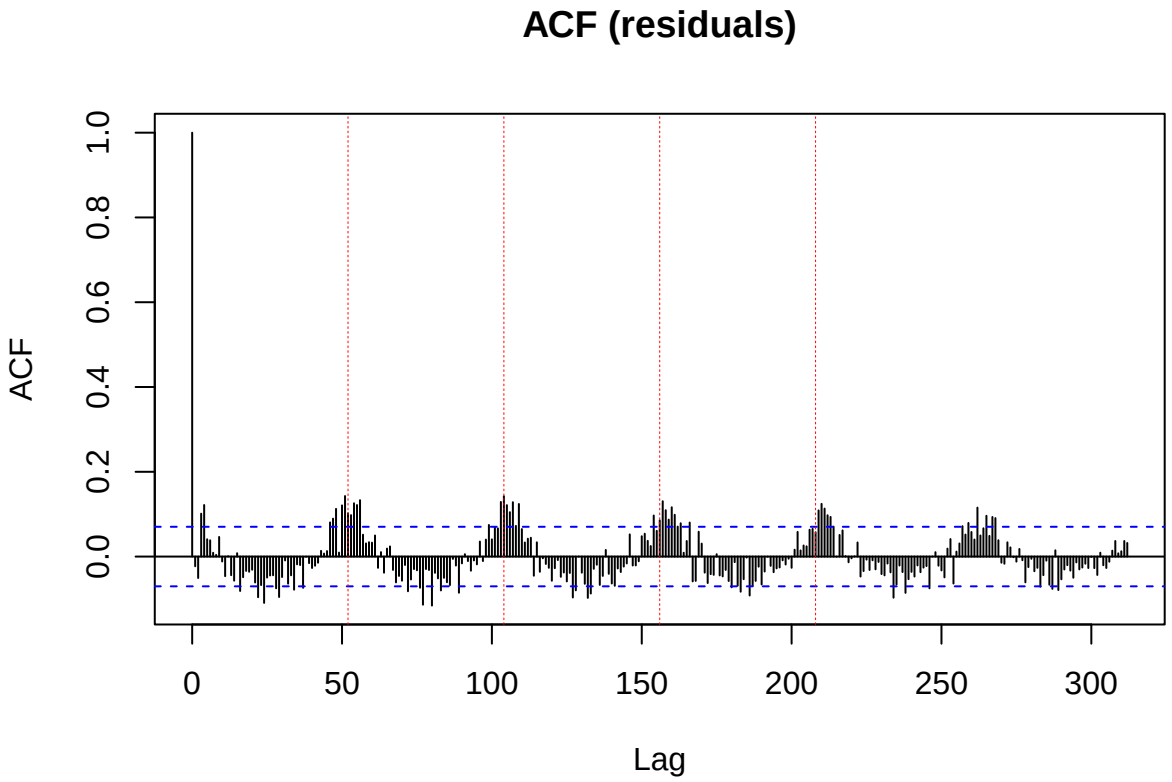
However, the predicted amplitude is smaller than the actual data. And the 95% CI is wide, indicating the uncertainty in the prediction.

5. (a) What are some of the pitfalls in using models to make predictions or forecasts of incidence in the future? Name at least two.

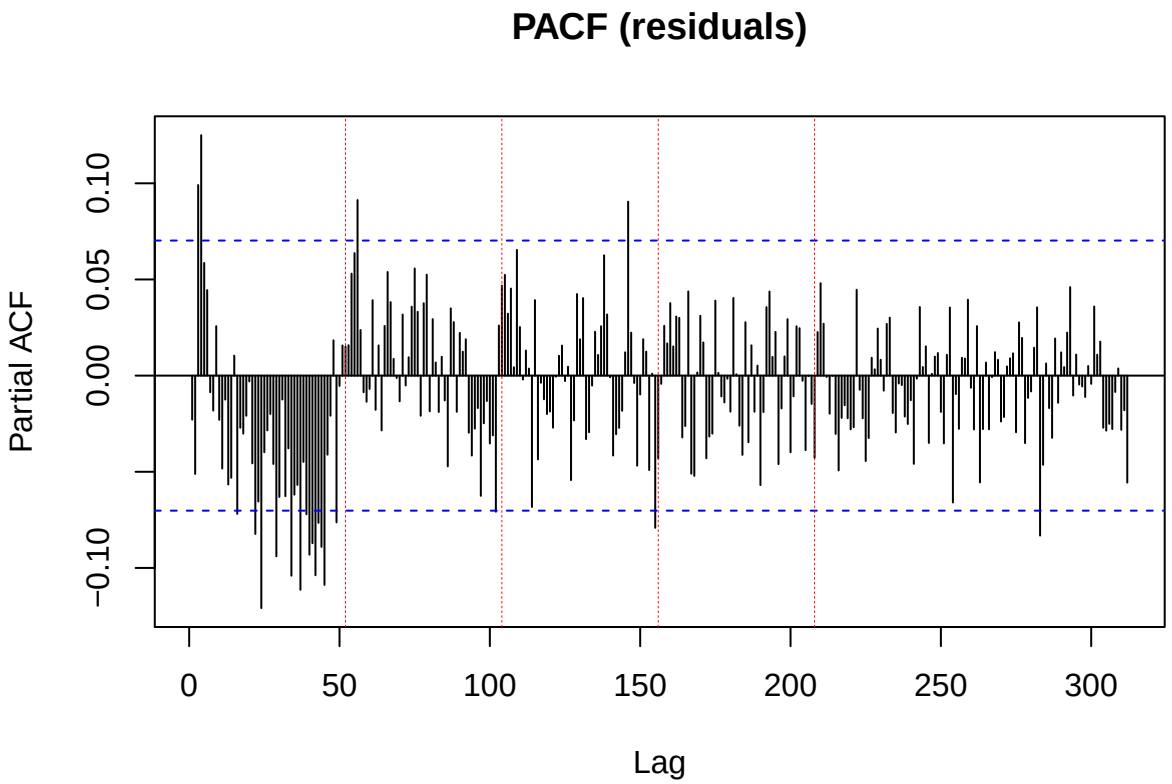
```
# Plot model residuals
par(mfrow=c(1,1))
plot(mod$residuals, type='l', col="blue", main="Model residuals",
     xlab="Time (in week)",ylab="Residuals")
abline(h=0, col="red")
abline(v=seq(52,52*20,by=52),col='black',lty=2) # some seasonality, period of 52
```




```
acf(mod$residuals,main='ACF (residuals)',lag.max=6*52)
abline(v=c(52,52*2,52*3,52*4), col="red", lty=2, lwd=0.3)
```




```
pacf(mod$residuals,main='PACF (residuals)',lag.max=6*52)
abline(v=c(52,52*2,52*3,52*4), col="red", lty=2, lwd=0.3)
```



Answer:

There are some seasonality.

Bars exceed/about to exceed the threshold in Week 52, 104, ... (yearly cycle).

5. (b) How do ARIMA models and TSIR models differ in their approach to making predictions/forecasts?

ARIMA models are fairly agnostic about the reasons behind the autocorrelation, trends, and "random shocks" present in the data

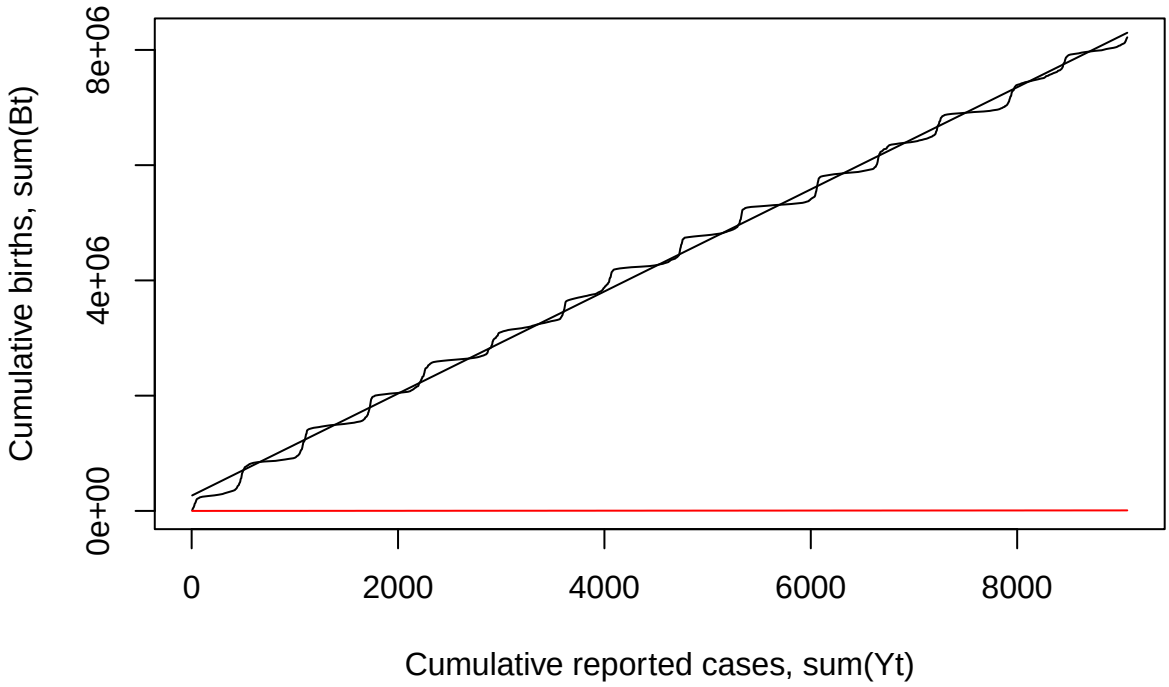
TSIR models explicitly attempt to attribute these to the infection process (i.e. interaction between susceptible and infectious individuals)

6. Assuming rotavirus has a generation interval of approximately 1 week and can be treated like an immunizing infection (and therefore can be modeled with a simple TSIR model), what is the reporting fraction, ρ, for rotavirus-positive laboratory tests from California? Why might this be such a small number?

```
cumreg <- lm(cumsum(rota$B) ~ cumsum(rota$rotaCA))
summary(cumreg)
```

```
##
## Call:
## lm(formula = cumsum(rota$B) ~ cumsum(rota$rotaCA))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -273406  -87752   -1245   89673  298188
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.646e+05  8.566e+03   30.89   <2e-16 ***
## cumsum(rota$rotaCA) 8.860e+02  1.661e+00  533.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 122400 on 778 degrees of freedom
## Multiple R-squared:  0.9973, Adjusted R-squared:  0.9973
## F-statistic: 2.844e+05 on 1 and 778 DF,  p-value: < 2.2e-16
```

```
plot(x=cumsum(rota$rotaCA),y=cumsum(rota$B),
     type='l',col='black',
     xlab='Cumulative reported cases, sum(Yt)',
     ylab='Cumulative births, sum(Bt)')
lines(x=cumsum(rota$rotaCA),y=cumreg$fitted.values,col='black')
lines(cumsum(rota$rotaCA), cumsum(rota$rotaCA),col="red") # <-- One-to-one line
```




```
# Under-reporting factor:
summary(cumreg)
```

```
##
## Call:
## lm(formula = cumsum(rota$B) ~ cumsum(rota$rotaCA))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -273406  -87752   -1245   89673  298188
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.646e+05  8.566e+03   30.89   <2e-16 ***
## cumsum(rota$rotaCA) 8.860e+02  1.661e+00  533.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 122400 on 778 degrees of freedom
## Multiple R-squared:  0.9973, Adjusted R-squared:  0.9973
## F-statistic: 2.844e+05 on 1 and 778 DF,  p-value: < 2.2e-16
```

```
coef(cumreg)[2] # This is 1/rho, so...
```

```
## cumsum(rota$rotaCA)
##             885.973
```

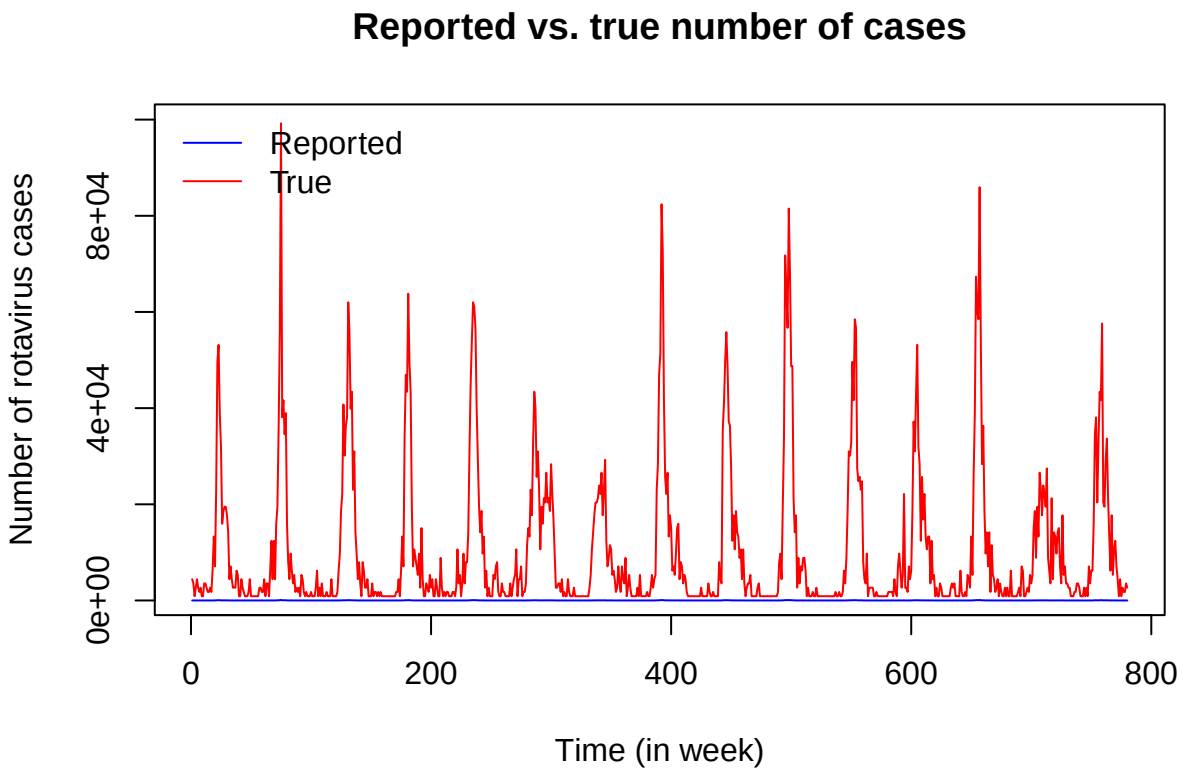
```
ur <- 1/coef(cumreg)[2] # this is rho ! :)
ur # 0.0011
```

```
## cumsum(rota$rotaCA)
##         0.001128703
```

```r
# True number of measles cases through time:
Ic <- (1/ur)*rota$rotaCA
head(Ic)
```

```
## [1] 4429.865 3543.892  885.973 2657.919 4429.865 2657.919
```

```r
# PLOT: Reported (Yt) vs. true number of cases (It)
plot(Ic, type='l', col="red", ylab="Number of rotavirus cases",
     xlab="Time (in week)",main="Reported vs. true number of cases")
lines(rota$rotaCA,col="blue") # This is reported cases
legend(x="topleft",legend=c("Reported","True"),
       col=c("blue","red"),
       lty=c(1,1),bty="n")
```



Answer:

The reporting fraction, ρ, is about 0.0011.

It is a small number because –

i. only a small number of cases are reported to the National Respiratory and Enteric Viral Surveillance System.

ii. the assumption that "rotavirus can be treated like an immunizing infection" does not hold (the immunity against rotavirus may decline over time). TSIR approach assumes infection is SIR, while rotavirus is best modeled by an SEIRS or SIRS model. Applying the TSIR susceptible reconstruction exaggerates the discrepancy between cumulative births and cumulative reported cases, producing a small estimate of the reporting fraction.

7. Estimate and plot β_u for each of the 52 weeks of the rotavirus season. (HINT: The population size of California in 2000 was 34 million. Your estimate of S_mean for the TSIR model will be between 1% and 50% of the total population.)

```
# This is just the residual, so...
D <- cumreg$residuals
```

```
# Create a new matrix "seas" with diag == 1 and else is 0.
# (This is to estimate j = 52 values of beta, rather than 546 beta's)

tf <- length(rota$B) # 780 (52 weeks x 15 years)
seas <- matrix(NA,779,52)

for (i in 1:(tf-1)) {
  for (j in 1:52){
    seas[i,j] = ifelse((rota$week[i])==j,1,0)
  }}
seas[1:10,1:10] # Big matrix with diag == 1 and else is 0
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    1    0    0    0    0    0    0    0    0     0
##  [2,]    0    1    0    0    0    0    0    0    0     0
##  [3,]    0    0    1    0    0    0    0    0    0     0
##  [4,]    0    0    0    1    0    0    0    0    0     0
##  [5,]    0    0    0    0    1    0    0    0    0     0
##  [6,]    0    0    0    0    0    1    0    0    0     0
##  [7,]    0    0    0    0    0    0    1    0    0     0
##  [8,]    0    0    0    0    0    0    0    1    0     0
##  [9,]    0    0    0    0    0    0    0    0    1     0
## [10,]    0    0    0    0    0    0    0    0    0     1
```

```
seas[,1] # 1 in every 52 entries. Week 1 in each year will have the same value of beta
```

```
##   [1] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
## [223] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [260] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [297] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [334] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
## [371] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [408] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [445] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
## [482] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [519] 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [556] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [593] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
## [630] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [667] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
## [704] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
## [741] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [778] 0 0
```

```
seas[,2]
```

```
##   [1] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
## [223] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [260] 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [297] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [334] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
## [371] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [408] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [445] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
## [482] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [519] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [556] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [593] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
## [630] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [667] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [704] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
## [741] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [778] 0 0
```

```
# Create lInew and lIold
lInew <- log(as.vector(Ic)[2:tf])
lIold <- log(as.vector(Ic)[1:(tf-1)])
head(cbind(lIold,lInew)) # 5.928401 prevalent cases transmitted to 6.337563 new cases at time 1
```

```
##         lIold    lInew
## [1,] 8.396124 8.172981
## [2,] 8.172981 6.786686
## [3,] 6.786686 7.885299
## [4,] 7.885299 8.396124
## [5,] 8.396124 7.885299
## [6,] 7.885299 7.479834
```

```
# Create Dold
Dold <- as.vector(D)[1:(tf-1)]

# Approximate size of london population
N <- 3300000

# We don't know how many of 3,300,000 were susceptible. Thus we have to consider
# various possibilities.
# Plausible values of S_mean is between 5-25% of the population size, so:
S_mean <- seq(0.05,0.25,0.001)*N

# Create an empty vector to store log-likelihoods
llik <- rep(0,length(S_mean))
```

```r
for (i in 1:length(S_mean)){
  # Calculate lSold
  lSold <- log(S_mean[i] + Dold) # S_t = S_mean + D_t (Slide 40)
  # Calculate the log likelihood for each value of S_mean by fitting a bunch of
  # linear regression models for different values of S_mean
  llik[i] <- logLik(glm(lInew ~ lIold + seas, offset = lSold))[1]
}
```

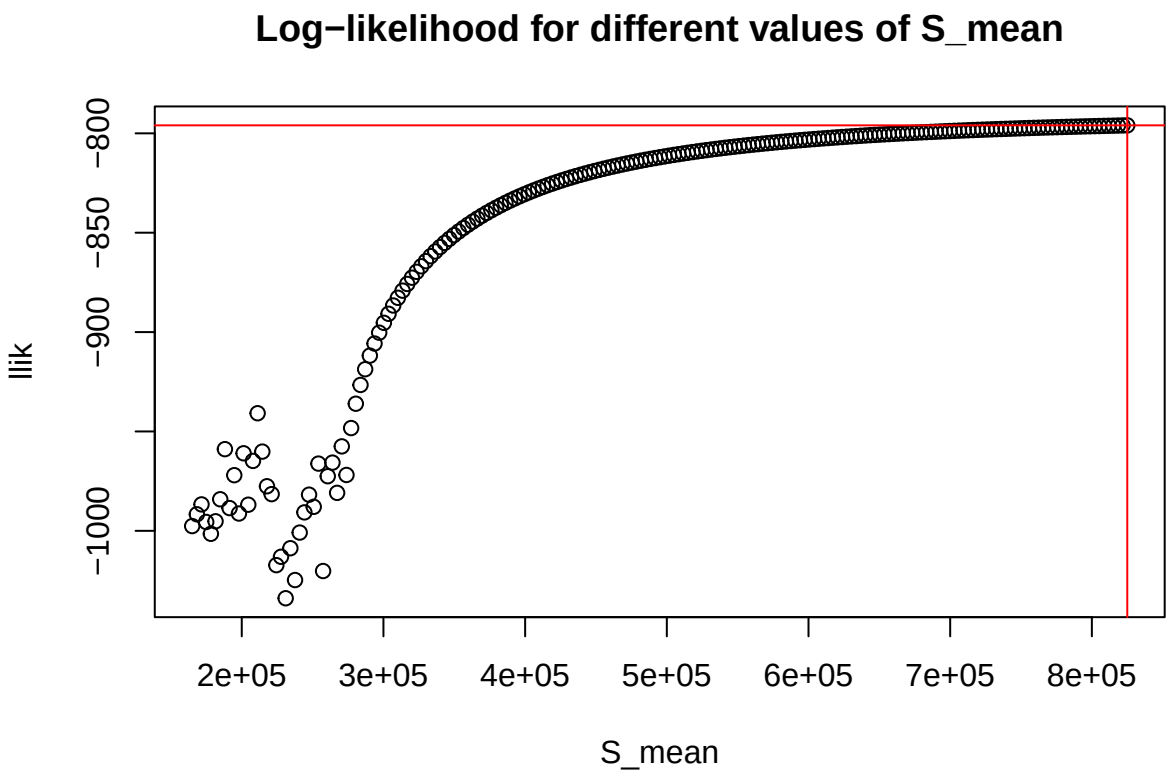
```
## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced
```

```
## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced

## Warning in log(S_mean[i] + Dold): NaNs produced
```

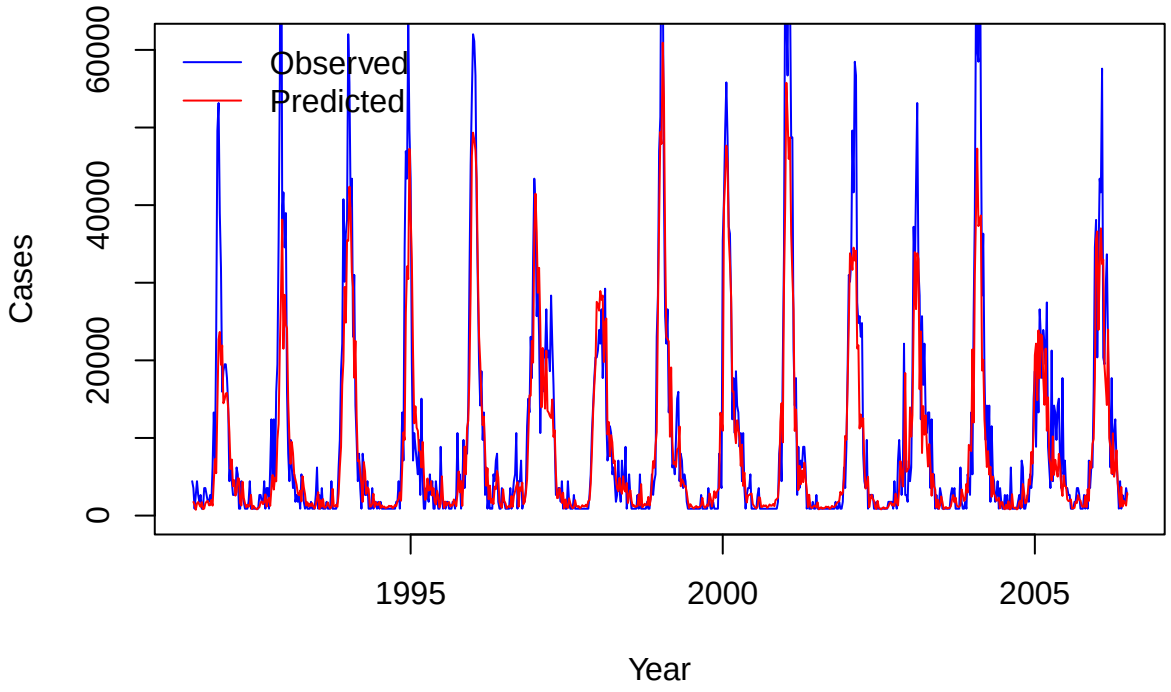
```r
# Let's take a look at the plot of likelihood
plot(x=S_mean,y=llik,main="Log-likelihood for different values of S_mean")

# The best fit is...
xi <- which.max(llik) # Returns index of the maximum log-likelihood estimate
Smean_estim <- S_mean[xi] # Smean value for the best-fit model
Smean_estim/N # About 10% of the population was susceptible to measles on average
```

```
## [1] 0.25
```

```r
abline(h=max(llik),col="red")
abline(v=Smean_estim,col="red")
```




```
# S_t for the best-fit model
lSold <- log(Smean_estim + Dold)

# Create the best-fitted model
bestmodfit <- glm(lInew ~ lIold + seas, offset = lSold)

# PLOT
plot(x = rota$time, y = (Ic), type='l',col='blue',
     ylim = c(0, max(exp(bestmodfit$fitted.values))),
     xlab='Year', ylab='Cases')
lines(x=rota$time[2:tf],
      y=exp(bestmodfit$fitted.values), # Need to take exp because everything is in the log scale
      col='red')
legend(x="topleft",legend=c("Observed","Predicted"),col=c("blue","red"),lty=c(1,1),bty="n")
```




```
coef(bestmodfit)
```

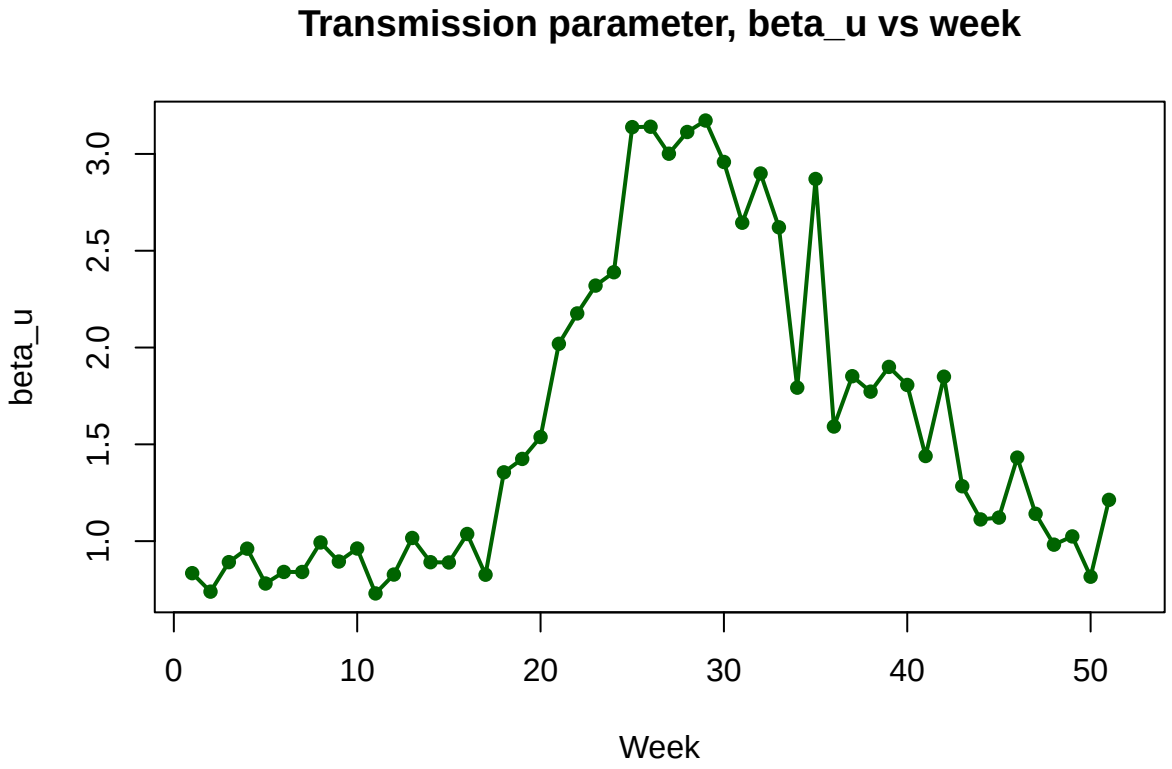
```
##   (Intercept)         llold         seas1         seas2         seas3
## -10.548494110   0.589701881  -0.180724344  -0.302183006  -0.114889424
##         seas4         seas5         seas6         seas7         seas8
##  -0.039577962  -0.246935693  -0.173295439  -0.173491187  -0.006793988
##         seas9        seas10        seas11        seas12        seas13
##  -0.111359411  -0.038809550  -0.314744349  -0.189437203   0.016358226
##        seas14        seas15        seas16        seas17        seas18
##  -0.115118322  -0.116645042   0.036332680  -0.190329951   0.304082620
##        seas19        seas20        seas21        seas22        seas23
##   0.353993397   0.430239875   0.702735709   0.777607221   0.841566932
##        seas24        seas25        seas26        seas27        seas28
##   0.870901938   1.143748198   1.144410055   1.098997590   1.135702787
##        seas29        seas30        seas31        seas32        seas33
##   1.154545977   1.084706399   0.972571037   1.064399345   0.963595045
##        seas34        seas35        seas36        seas37        seas38
##   0.583741031   1.054548141   0.465046310   0.616269951   0.572252522
##        seas39        seas40        seas41        seas42        seas43
##   0.641903827   0.591523015   0.364399727   0.614890319   0.249638177
##        seas44        seas45        seas46        seas47        seas48
##   0.106362841   0.114873903   0.358660004   0.132151318  -0.018324001
##        seas49        seas50        seas51        seas52
##   0.024186734  -0.202977860   0.193332091            NA
```

```
coef(bestmodfit) # estimated beta's are stored in 3 - 54
```

```
##   (Intercept)         llold         seas1         seas2         seas3
## -10.548494110   0.589701881  -0.180724344  -0.302183006  -0.114889424
##         seas4         seas5         seas6         seas7         seas8
##  -0.039577962  -0.246935693  -0.173295439  -0.173491187  -0.006793988
##         seas9        seas10        seas11        seas12        seas13
##  -0.111359411  -0.038809550  -0.314744349  -0.189437203   0.016358226
```

```
##        seas14        seas15        seas16        seas17        seas18
##  -0.115118322  -0.116645042   0.036332680  -0.190329951   0.304082620
##        seas19        seas20        seas21        seas22        seas23
##   0.353993397   0.430239875   0.702735709   0.777607221   0.841566932
##        seas24        seas25        seas26        seas27        seas28
##   0.870901938   1.143748198   1.144410055   1.098997590   1.135702787
##        seas29        seas30        seas31        seas32        seas33
##   1.154545977   1.084706399   0.972571037   1.064399345   0.963595045
##        seas34        seas35        seas36        seas37        seas38
##   0.583741031   1.054548141   0.465046310   0.616269951   0.572252522
##        seas39        seas40        seas41        seas42        seas43
##   0.641903827   0.591523015   0.364399727   0.614890319   0.249638177
##        seas44        seas45        seas46        seas47        seas48
##   0.106362841   0.114873903   0.358660004   0.132151318  -0.018324001
##        seas49        seas50        seas51        seas52
##   0.024186734  -0.202977860   0.193332091            NA
```

```r
# Plot the values of beta_u vs week
plot(x=1:52, y=exp(coef(bestmodfit)[3:54]),
     col="darkgreen",lwd=2,type='o',pch=16,xlab='Week',ylab='beta_u',
     main="Transmission parameter, beta_u vs week")
```




```r
# This is the seasonal pattern of transmission in our best-fitted model

# From here, you can estimate relationships between beta and various factors
# (e.g., school terms, temperature, rain fall...). This is the advantage of
# TSIR, compared to ARIMA.

# What is the value of alpha?
alpha <- coef(bestmodfit)[2]
```

```
alpha # Close to 1, which suggests ...
```

```
##     lIold
## 0.5897019
```

8. Where does the transmission rate (β_u) peak relative to the peak in the number of rotavirus cases (a) in 1991-92 (i.e. the first rotavirus season) and (b) 2005-2006 (i.e. the last rotavirus season)? What might explain the difference?

i. Identify the peak

```
beta_u <- exp(coef(bestmodfit)[3:54])
peak_beta_week <- which.max(beta_u)
peak_beta_week   # 29
```

```
## seas29
##     29
```

ii. The relative positions of the peak in 1991–92 and 2005–06 with respect to the 29th week

```
weeks_per_season <- 52
n_seasons <- length(rota$rotaCA) / weeks_per_season  # 15

# The first season: 1991-92
idx_1991 <- 1:weeks_per_season

# The last season: 2005-06
idx_2005 <- ((n_seasons - 1) * weeks_per_season + 1):(n_seasons * weeks_per_season)

# The week numbers within each season corresponding to the peak number of cases
peak_case_week_1991 <- rota$week[idx_1991][which.max(rota$rotaCA[idx_1991])]
peak_case_week_2005 <- rota$week[idx_2005][which.max(rota$rotaCA[idx_2005])]

# The relative difference compared to the peak value of beta_u (in week 29)
lag_1991 <- peak_case_week_1991 - peak_beta_week
lag_2005 <- peak_case_week_2005 - peak_beta_week

cat("The transmission rate peaks at:  Week", peak_beta_week, "\n")
```

```
## The transmission rate peaks at:  Week 29
```

```
cat("1991-92 rotavirus cases peak at: Week", peak_case_week_1991,
    " (lag =", lag_1991, "weeks)\n")
```

```
## 1991-92 rotavirus cases peak at: Week 23  (lag = -6 weeks)
```

```
cat("2005-06 rotavirus cases peak at: Week", peak_case_week_2005,
    " (lag =", lag_2005, "weeks)\n")
```

```
## 2005-06 rotavirus cases peak at: Week 31  (lag = 2 weeks)
```

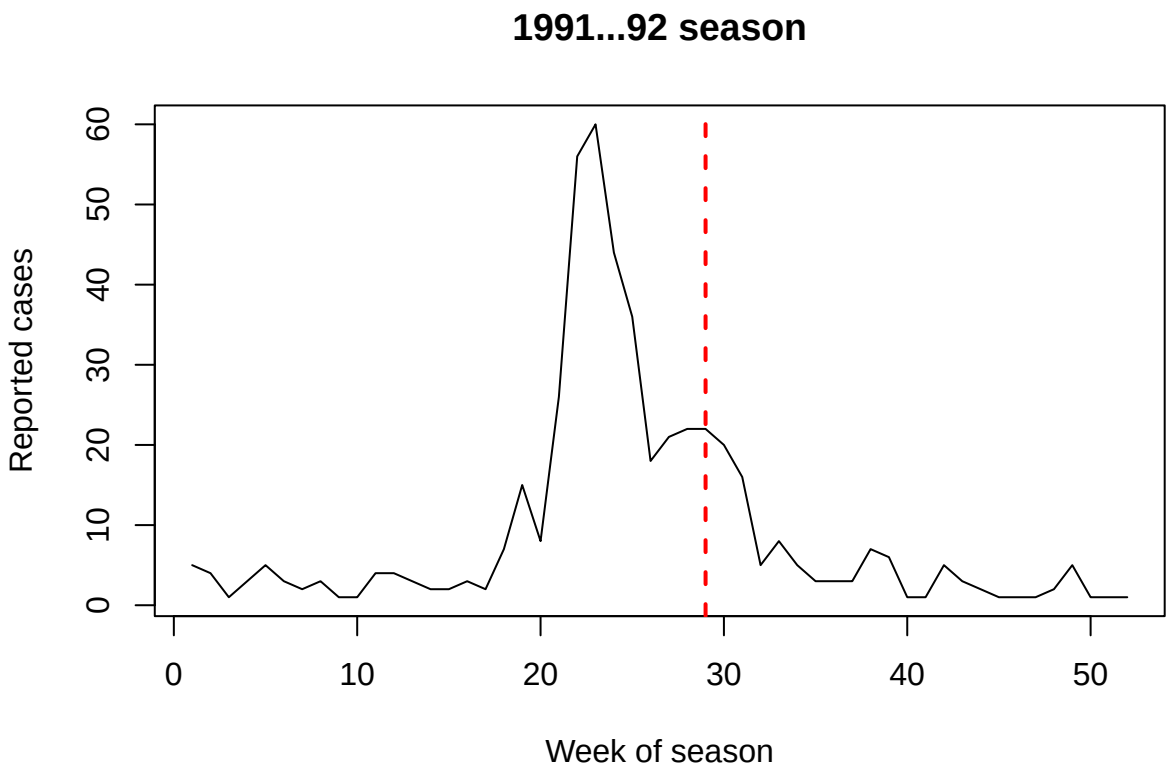
```r
# Plot 1991-92 season
plot(rota$week[idx_1991], rota$rotaCA[idx_1991], type = "l",
     xlab = "Week of season", ylab = "Reported cases",
     main = "1991-92 season")
```

```
## Warning in title(...): conversion failure on '1991-92 season' in 'mbcsToSbcs':
## dot substituted for <e2>
```

```
## Warning in title(...): conversion failure on '1991-92 season' in 'mbcsToSbcs':
## dot substituted for <80>
```

```
## Warning in title(...): conversion failure on '1991-92 season' in 'mbcsToSbcs':
## dot substituted for <93>
```

```r
abline(v = peak_beta_week, col = "red", lty = 2, lwd = 2)
```



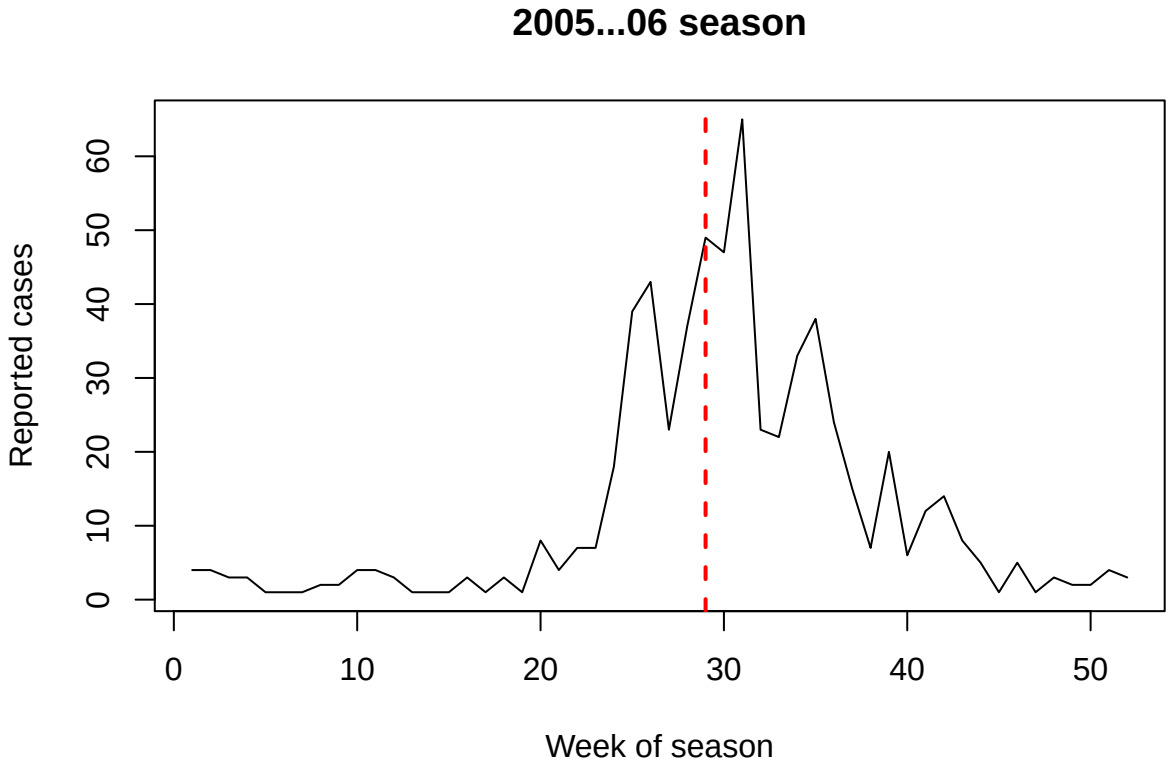

```r
## Plot 2005-06 season
plot(rota$week[idx_2005], rota$rotaCA[idx_2005], type = "l",
     xlab = "Week of season", ylab = "Reported cases",
     main = "2005-06 season")
```

```
## Warning in title(...): conversion failure on '2005-06 season' in 'mbcsToSbcs':
## dot substituted for <e2>
```

```
## Warning in title(...): conversion failure on '2005-06 season' in 'mbcsToSbcs':
## dot substituted for <80>
```

```
## Warning in title(...): conversion failure on '2005-06 season' in 'mbcsToSbcs':
## dot substituted for <93>
```

```r
abline(v = peak_beta_week, col = "red", lty = 2, lwd = 2)
```



Answer:

In the TSIR model, the transmission rate (β_u) peaks around Week 29.

- In 1991-92 (i.e. the first rotavirus season), the number of rotavirus cases peaks around Week 23, about 6 weeks before the transmission rate (β_u) peak.
- In 2005-2006 (i.e. the last rotavirus season), the number of rotavirus cases peaks around Week 31, about 2 weeks after the transmission rate (β_u) peak.

The number of new infections at time $t+1$ is

$$I_{t+1} \propto \beta_t S_t I_t^{\alpha}$$

- where S_t and I_t are the number of susceptible and infectious individuals, respectively, in the previous generation
- β_t is the transmission rate at time t
- α is a correction factor for non-homogeneous mixing

Since the case peak is determined by both the transmission rate β_t and the number of susceptible (S_t),

- if in a certain year, the number of susceptible (S_t) is high, once β slightly increases, the number of infections (I_t) will rapidly surge, and the peak of cases will occur earlier than the peak of the transmission rate.
- if in a certain year, the number of susceptible (S_t) is low, even if β increases, the number of infections (I_t) is hard to increase, and the peak of cases will occur later than the peak of the transmission rate.

Therefore, the difference between the transmission rate peak relative to the peak in the number of rotavirus cases in 1991-92 and that in 2005-2006 might be caused by the fact that the prevalence of the disease in the past years led to a decrease in the number of susceptible individuals.