analisys_modena_cancer

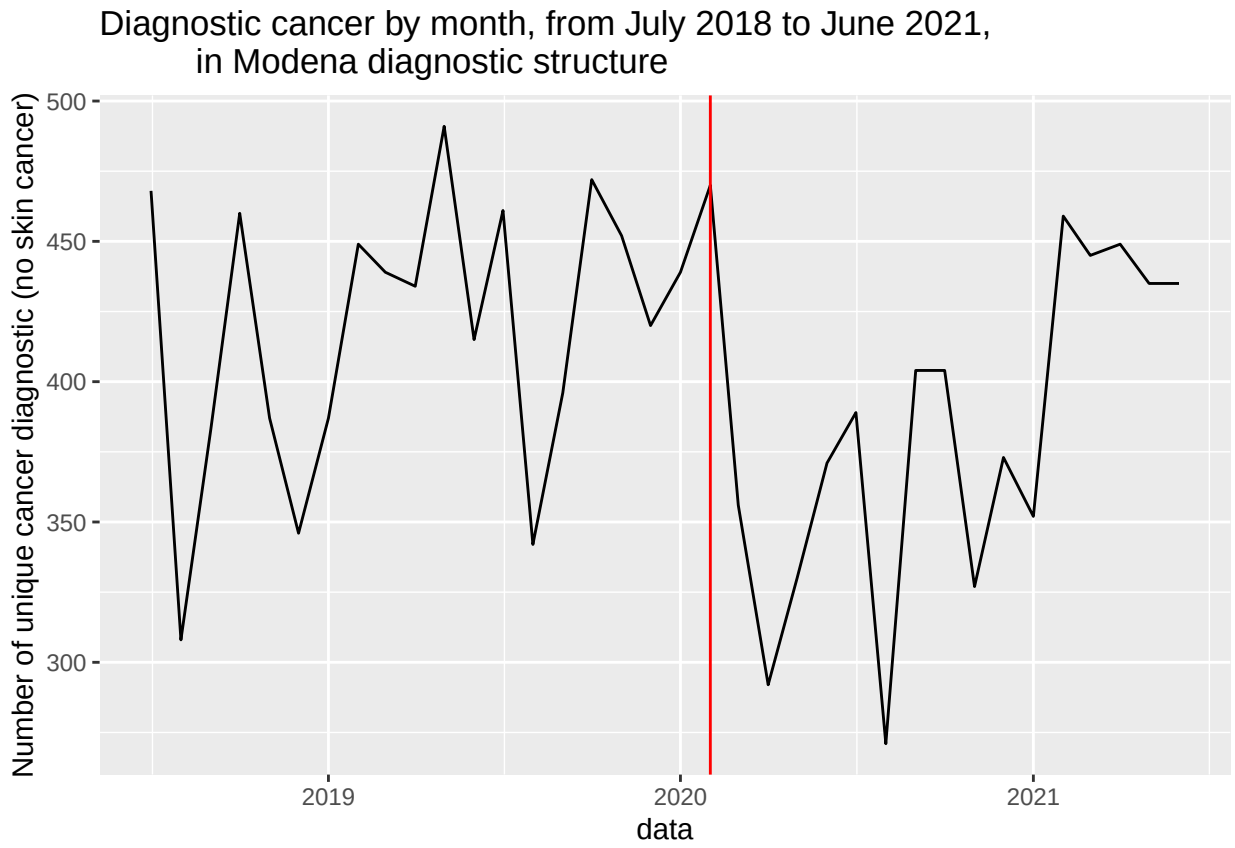
Rmarkdwons settings

Loading the dataset of month incidence

Exploratory data analysis:

```
# lettura dei dati come una serie storiche, specificando il periodo di tempo
ts_tot = ts(dati_all$conteggio,start = c(2018,7), end=c(2021,6), frequency = 12)
time=seq(as.Date("2018-7-1"), as.Date("2021-6-1"), by = "months")
df_tot = as.data.frame(ts_tot)
# Generazione del grafico
ggplot(df_tot, aes(time)) + # plot dei dati, linea per delineare lo scoppio della pandemia
  ggtitle("Diagnostic cancer by month, from July 2018 to June 2021,
          in Modena diagnostic structure") +
  geom_line(aes(y = ts_tot)) +
  geom_vline(xintercept = as.numeric(time[20]), color="red") + # barra rossa verticale
  xlab("data") +
  ylab("Number of unique cancer diagnostic (no skin cancer)")
```

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



Dataframe of 12 months before and 12 anfter the covid pandemic

```
# 12 months dataser
# horizontal datafrmae
pre = dati_all$conteggio[8:19]
post = dati_all$conteggio[20:31]
# two columns, one before and one after covid
df_1920_12 = as.data.frame(t(rbind(pre,post)))
colnames(df_1920_12) = c("before","after") # columns names
#### Vertical Dataframae
df_1920v_12 = as.data.frame(matrix(data=c(df_1920_12[,1],df_1920_12[,2],rep(0,12),rep(1,12)),
                                   nrow = 24,ncol = 2, byrow =F))
colnames(df_1920v_12) =c("conteggio","periodo")
df_1920v_12$periodo = as.factor(df_1920v_12$periodo)
```

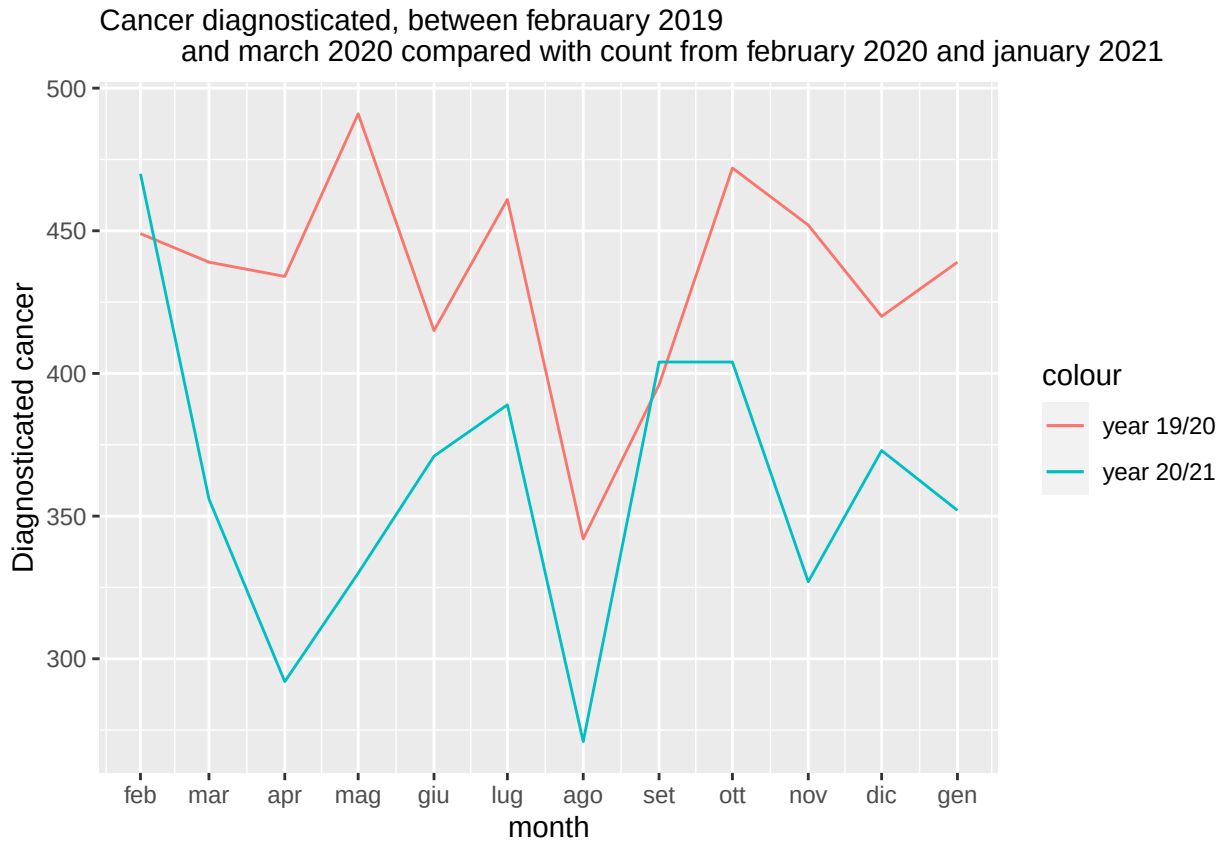
Second dataframe with 5 months and the correspondent 5 months in 2020 and 2019 (pre-covid)

```
# last 5 month
# horizontal datafrmae
pre = dati_all$conteggio[8:12]
post = dati_all$conteggio[20:24]
post2 = dati_all$conteggio[32:36]

df_1920_5 = as.data.frame(t(rbind(pre,post,post2)))
colnames(df_1920_5) = c("before","after","last5") # columns names
#### Vertical dataframe
df_1920v_5 = as.data.frame(matrix(data=c(pre,post,post2,rep(0,5),rep(1,5),rep(2,5)),
                                   nrow = 15,ncol = 3, byrow =F))
colnames(df_1920v_5) =c("conteggio","periodo") # columns names
df_1920v_5$periodo = as.factor(df_1920v_5$periodo)
```

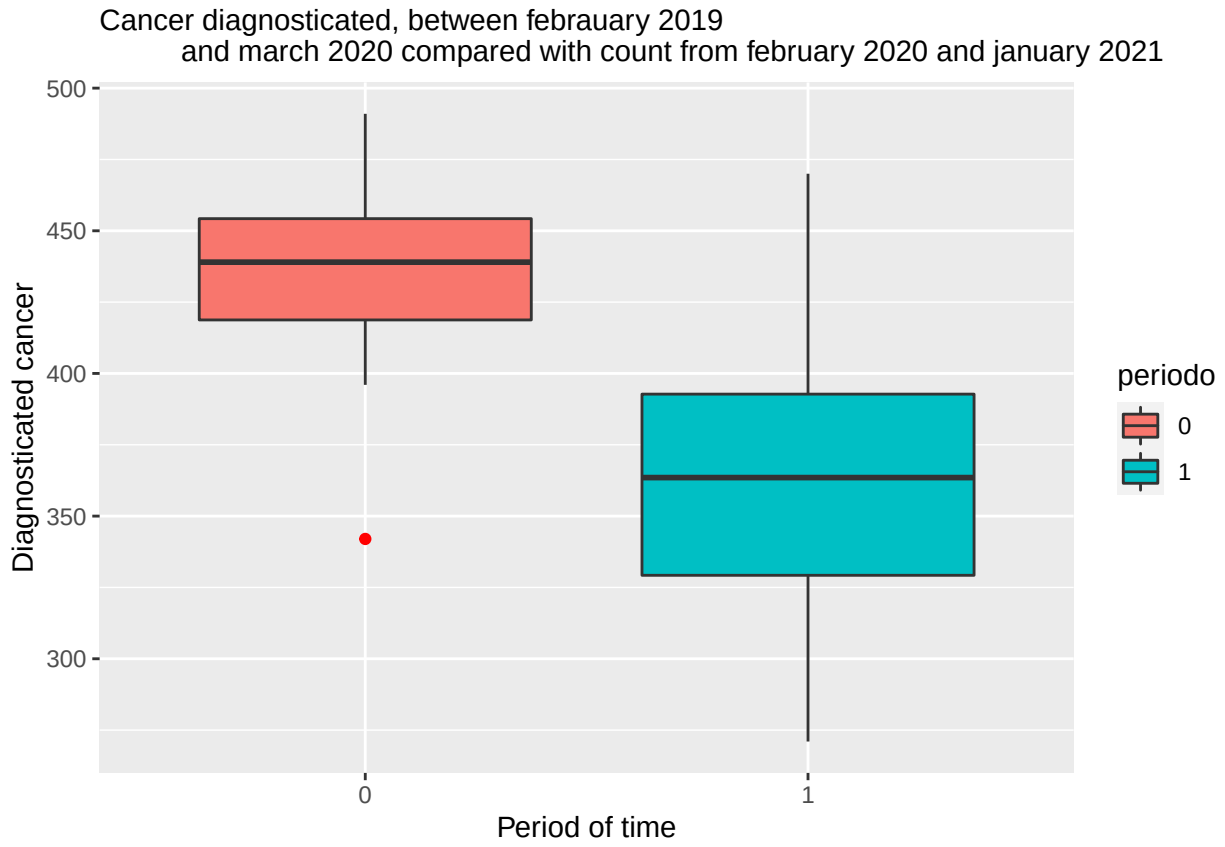
Plot 2 - 12 months comparasion

```
time_mesi=seq(as.Date("2019-2-1"), as.Date("2020-1-1"), by = "months")
ggplot(df_1920_12, aes(time_mesi)) +
  ggtitle("Cancer diagnosticated, between febrauary 2019
          and march 2020 compared with count from february 2020 and january 2021") +
  geom_line(aes(y = before, colour = "year 19/20")) +
  geom_line(aes(y = after, colour = "year 20/21")) +
  xlab("month") +
  ylab("Diagnosticated cancer") +
  scale_x_date(breaks = '1 month',
               date_labels = "%b") +
  theme(plot.title = element_text(size = 11))
```



Plot 3 - Boxplot comparasion

```
ggplot(df_1920v_12, aes(x=periodo,y=conteggio,fill=periodo)) + # boxplot dei 12 mesi pre covid - ROSSO
         geom_boxplot(outlier.colour = "red",) +
         ggtitle("Cancer diagnosticated, between febrauary 2019
          and march 2020 compared with count from february 2020 and january 2021") +
         xlab("Period of time") +
         ylab("Diagnosticated cancer") +
         theme(plot.title = element_text(size = 11))
```



Mean difference observed

```
n1_12 = sum(df_1920_12$before);n1_12 # numerosità nei 12 mesi prima
```

```
## [1] 5210
```

```
n2_12 = sum(df_1920_12$after);n2_12 # numerosità nei 12 mesi after
```

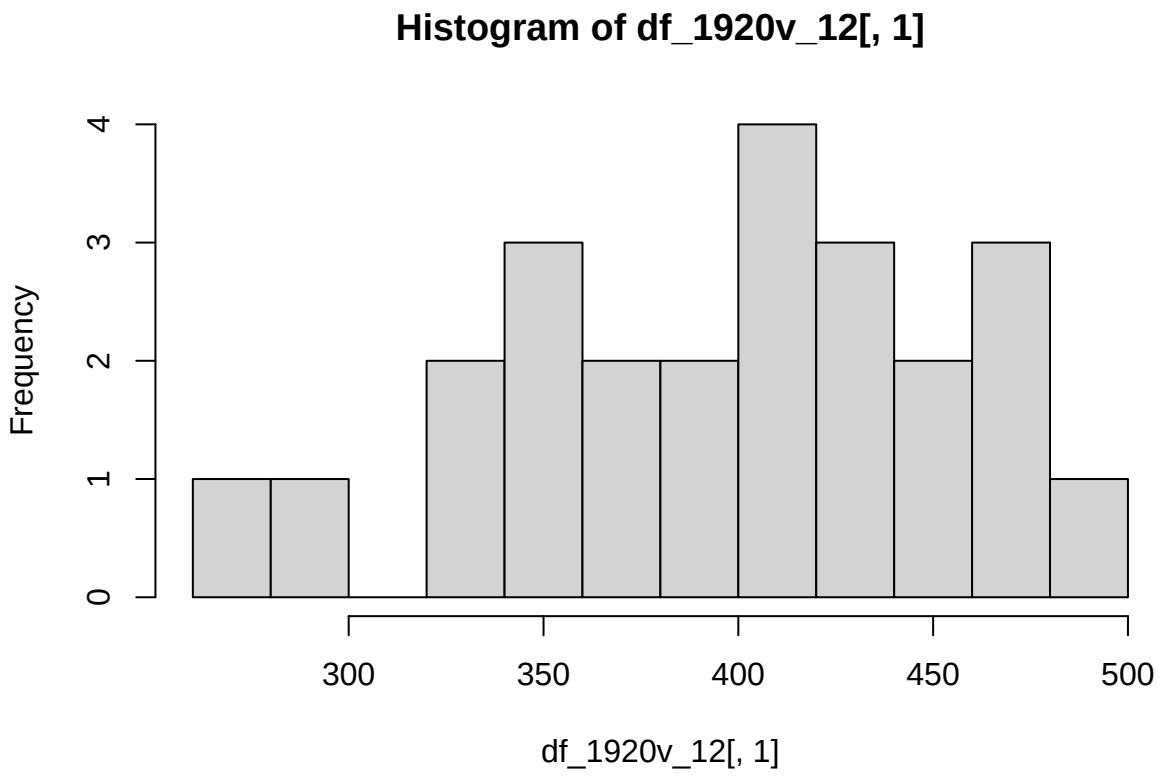
```
## [1] 4339
```

```
1-(n2_12/n1_12) # rapporto che denota un calo del 16.7%
```

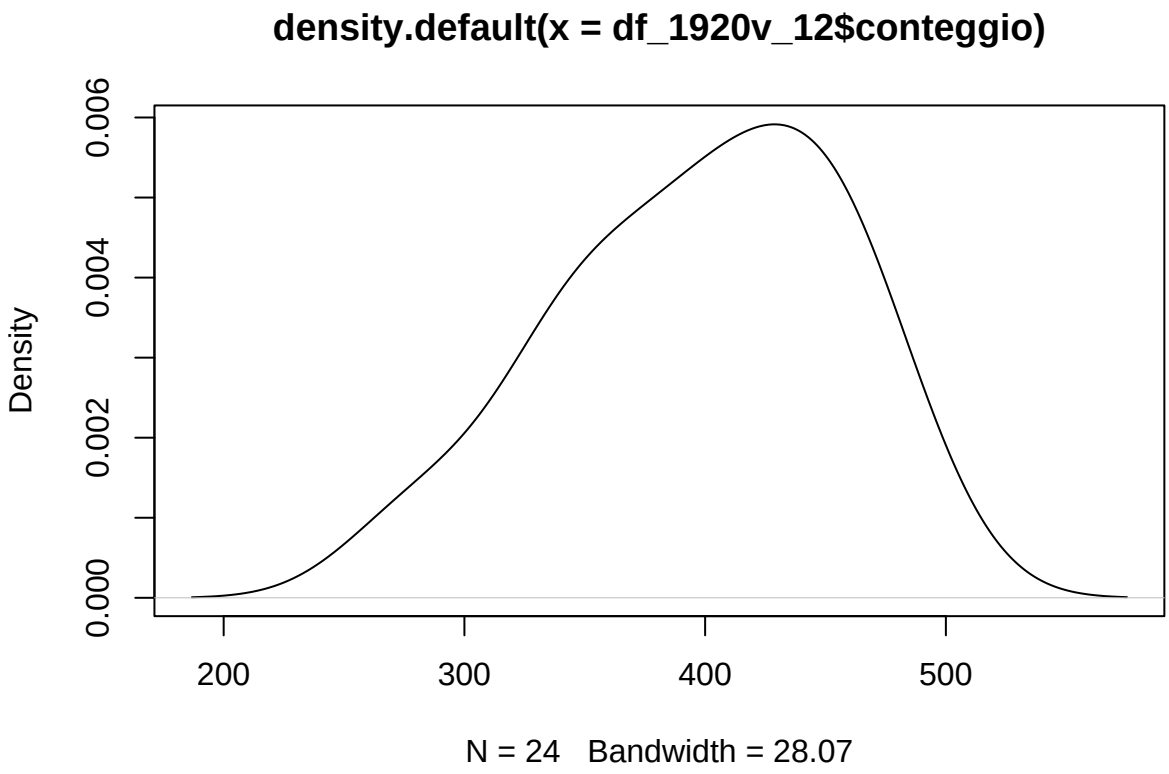
```
## [1] 0.1671785
```

Data distribution analysis

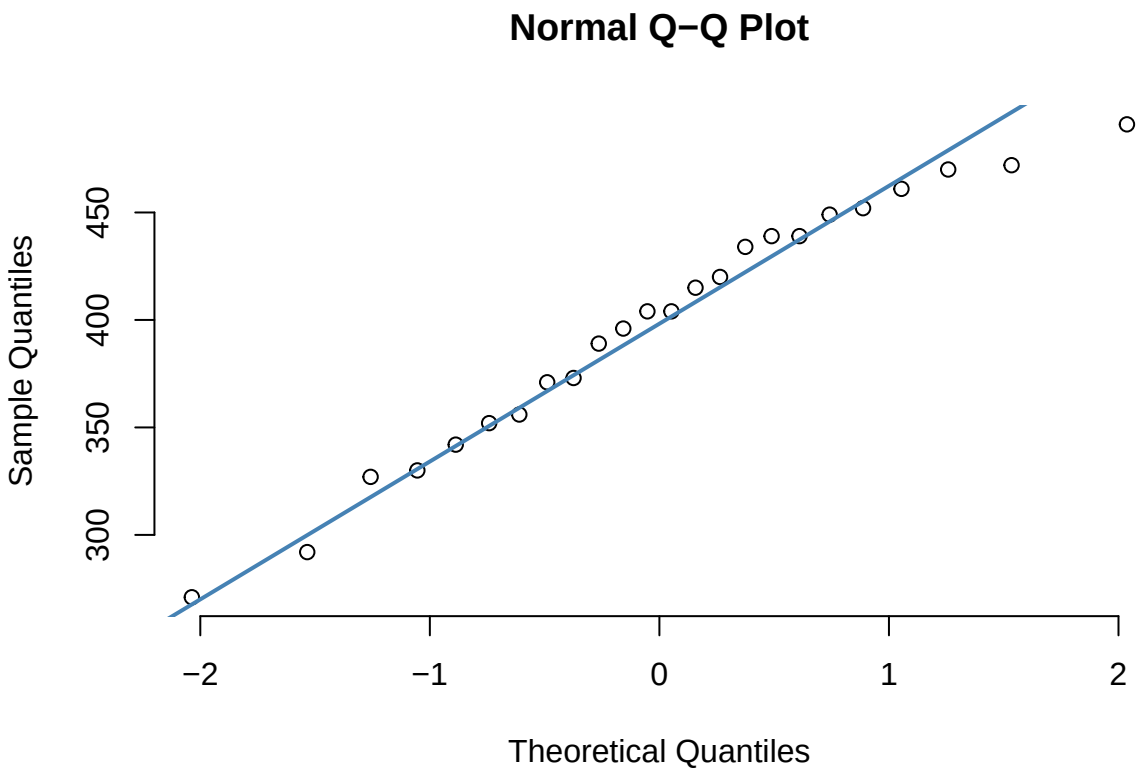
```
hist(df_1920v_12[,1],breaks = 8)
```




```
# Kernel Density Plot
d <- density(df_1920v_12$conteggio) # returns the density data
plot(d) # plots the results
```




```
qqnorm(df_1920v_12$conteggio, pch = 1, frame = FALSE)
qqline(df_1920v_12$conteggio, col = "steelblue", lwd = 2)
```




```
# qqplot(df_1920v$conteggio)
shapiro.test(df_1920v_12$conteggio)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_1920v_12$conteggio
## W = 0.96812, p-value = 0.6208
```

```
shapiro.test(df_1920_12$before)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_1920_12$before
## W = 0.93423, p-value = 0.4271
```

```
shapiro.test(df_1920_12$after)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_1920_12$after
## W = 0.97775, p-value = 0.9731
```

Normality can be assumed but the nature of the data and the dimension of the sample suggest the try a distribution for positive counting data like Poisson or Negative Binomial.

In our case Poisson can't be used because the variance is much bigger compared to the mean of the distribution.

(violation of one of the assumption for the Poisson distribution)

Checking mean and variance difference

```
mean(df_1920_12$before)
```

```
## [1] 434.1667
```

```
mean(df_1920_12$after)
```

```
## [1] 361.5833
```

```
var(df_1920_12)
```

```
##          before    after
## before 1498.697  714.803
## after   714.803 2880.629
```

Test for equal variance in both the sample

```
bartlett.test(df_1920v_12$conteggio~df_1920v_12$periodo)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  df_1920v_12$conteggio by df_1920v_12$periodo
## Bartlett's K-squared = 1.1036, df = 1, p-value = 0.2935
```

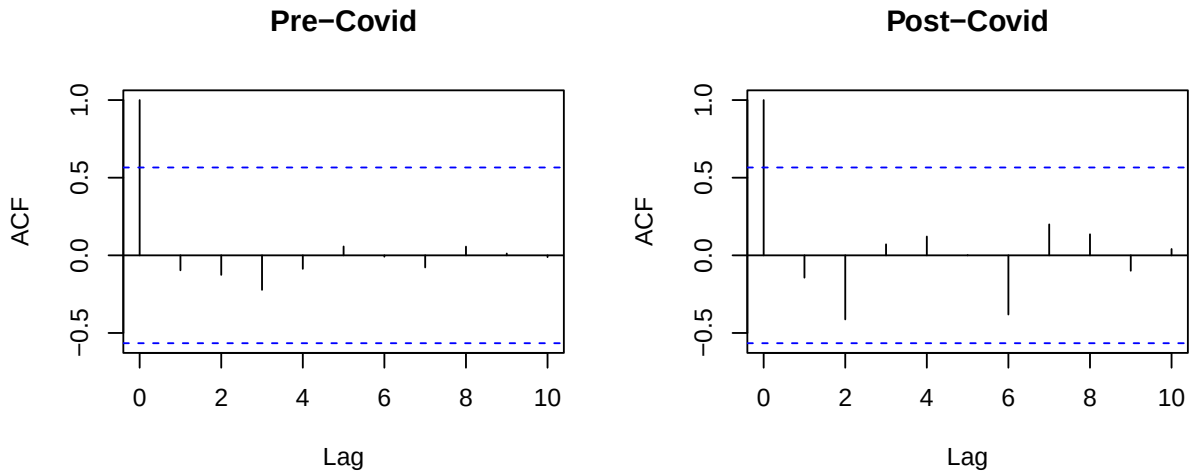
Confirmed the equal variance.

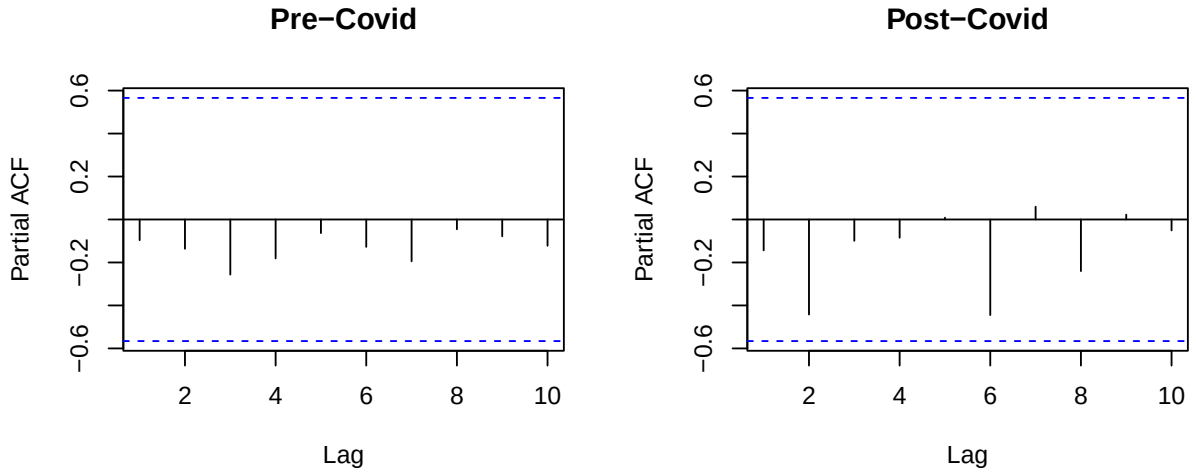
Now watch the autocorrelation function and partial autocorrelation function for check the independence assumption of our observation.

```
par(mfrow=c(2,2)) # 2 disegni in un riquadro

plt_ts_1 = acf(ts(df_1920_12[,1]), plot=F)
plot(plt_ts_1, main="Pre-Covid")
plt_ts_2 = acf(ts(df_1920_12[,2]), plot=F)
plot(plt_ts_2, main="Post-Covid")

plt_ts_1 = pacf(ts(df_1920_12[,1]), plot=F)
plot(plt_ts_1, main="Pre-Covid")
plt_ts_2 = pacf(ts(df_1920_12[,2]), plot=F)
plot(plt_ts_2, main="Post-Covid")
```





No big signal of correlation between observation in different times, we know about august seasonality.

Dummy variables for model the strong seasonality of august

```
# dummy
dm=make.dummy(length(df_1920v_12[,1]),start=2, freq=12)
# dm[1:6,]
# give names to the dummies
nomi=c("dm4","dm5","dm6","dm7","dm8","dm9","dm10","dm11","dm12","dm1","dm2","dm3")
dimnames(dm)=list(NULL,nomi)
colnames(dm)<-nomi
```

Linear model with normal assumption, t-test and Wilconox-Mann test

```
# 12 month
# Linear model
summary(lm(df_1920v_12$conteggio~df_1920v_12$periodo+dm[,8]))
```

```
##
## Call:
## lm(formula = df_1920v_12$conteggio ~ df_1920v_12$periodo + dm[,
##     8])
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -77.89 -19.04   0.00  18.67 100.11
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            442.47      11.15  39.691  < 2e-16 ***
## df_1920v_12$periodo1   -72.58      15.42  -4.707  0.00012 ***
## dm[, 8]                -99.68      27.89  -3.574  0.00179 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.77 on 21 degrees of freedom
## Multiple R-squared:  0.6245, Adjusted R-squared:  0.5888
## F-statistic: 17.47 on 2 and 21 DF,  p-value: 3.412e-05
```

```
# T test:
t.test(df_1920v_12$conteggio~df_1920v_12$periodo, var.equal =T)
```

```
##
##  Two Sample t-test
##
## data:  df_1920v_12$conteggio by df_1920v_12$periodo
## t = 3.7995, df = 22, p-value = 0.0009823
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##   32.96509 112.20157
## sample estimates:
## mean in group 0 mean in group 1
##        434.1667        361.5833
```

```
# Wilcoxon-Mann-Whitney test:
wilcox.test(df_1920v_12$conteggio~df_1920v_12$periodo)
```

```
## Warning in wilcox.test.default(x = c(449, 439, 434, 491, 415, 461, 342, : cannot
## compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_1920v_12$conteggio by df_1920v_12$periodo
## W = 125, p-value = 0.002426
## alternative hypothesis: true location shift is not equal to 0
```

Test with negative binomial assumption for overdispersed counting data

```
# 12 month
bn_covid = glm.nb(df_1920v_12$conteggio~df_1920v_12$periodo+dm[,8])
summary(bn_covid)
```

```
##
## Call:
## glm.nb(formula = df_1920v_12$conteggio ~ df_1920v_12$periodo +
##     dm[, 8], init.theta = 175.0351894, link = log)
##
## Deviance Residuals:
```

```
##      Min        1Q    Median        3Q       Max
## -2.42351  -0.53273  -0.02723   0.47449   2.77921
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           6.09452    0.02641 230.765  < 2e-16 ***
## df_1920v_12$periodo1 -0.18359    0.03710  -4.949 7.47e-07 ***
## dm[, 8]              -0.28314    0.06979  -4.057 4.98e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(175.0352) family taken to be 1)
##
##     Null deviance: 64.633  on 23  degrees of freedom
## Residual deviance: 24.213  on 21  degrees of freedom
## AIC: 248.08
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  175.0
##           Std. Err.:  73.6
##
##  2 x log-likelihood:  -240.077
```

```
# percentage difference after covid can be calculated with 1-exp(-0.18359)
1-exp(-0.18359)
```

```
## [1] 0.167723
```

Check for correct assumption of Negative binomial instead of Poisson

```
poi_covid <- glm(df_1920v_12$conteggio ~ df_1920v_12$periodo, family = "poisson")
pchisq(2 * (logLik(bn_covid) - logLik(poi_covid)), df = 1, lower.tail = FALSE)
```

```
## 'log Lik.' 6.578825e-18 (df=4)
```

confirm the correct assumption of negative binomial instead of Poisson.

Both models confirm the mean difference between the two sample. Same conclusion.