

Avaliação 2

Gabriel de Jesus Pereira - 20200121424

Pacotes utilizados

```
library(seastests)
library(astsa)
library(forecast)
library(modeltime)
library(tidyverse)
library(tseries)
library(timetk)
library(lmtest)
library(feasts)
library(tidymodels)
library(rsample)
```

Carregando a série

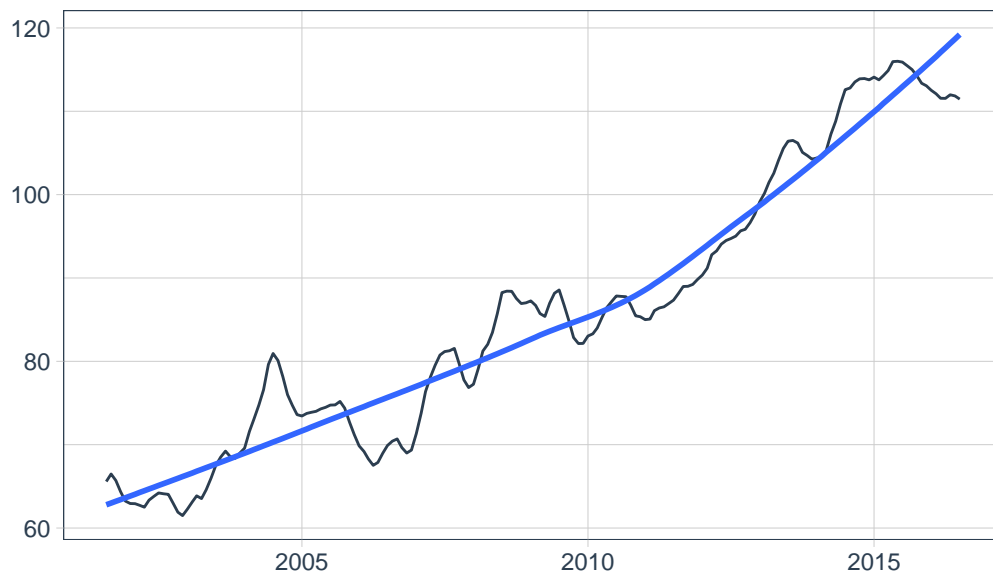
```
data <- tibble(
  chic = as.matrix(chicken) |> as.vector(),
  date = zoo::as.Date(time(chicken))
)
time_splits <- initial_time_split(data, prop = 0.8)
train <- training(time_splits)
test <- testing(time_splits)
```

Apresente a análise descritiva da série. Comente os resultados

Gráfico da série temporal

Analisando somente o gráfico da série, podemos ver que ela aparenta ter tendência. Sobre a sazonalidade, o preço das aves inteiras parecem cair sempre aos finais de ano, tendo uma queda abrupta no ano de 2006. A série aparenta ser sazonal, mas tem uma parece ter uma sazonalidade bastante pequena.

```
data |>
  plot_time_series(
    .date_var = date,
    .value = chic,
    .interactive = FALSE,
    .title = "")
```

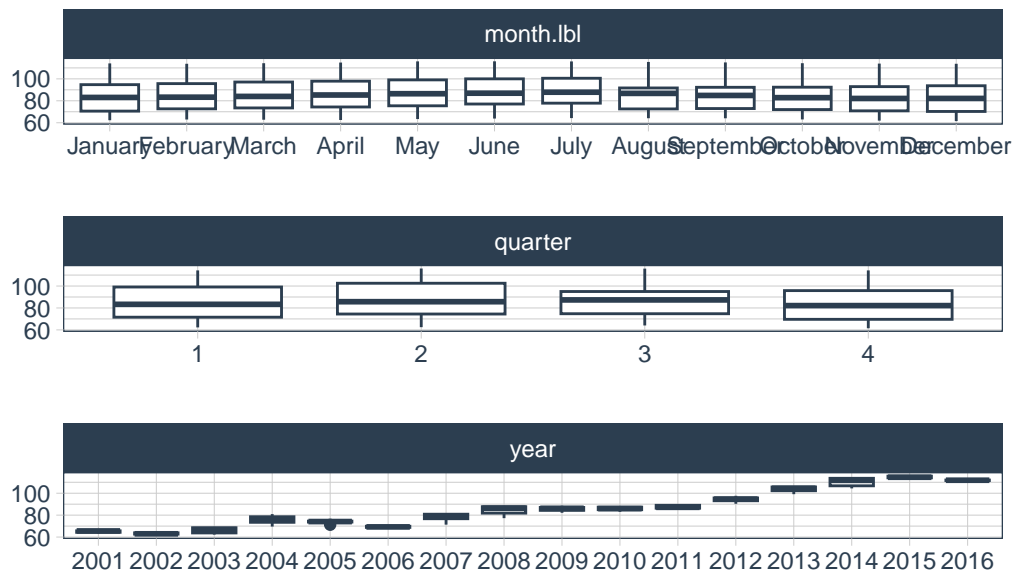


Comportamento sazonal da série

Como foi visto no gráfico da série, o preço das aves passam a cair próximo do final do ano. Pelo gráfico de boxplot dos meses, vemos que isso começa a acontecer a partir de agosto e vai até dezembro. De janeiro a junho o preço das aves são bastante parecidas mas tendem

a crescer. Além disso, podemos ver que a partir do terceiro trimestre é o momento em que o preço das aves mais caem. Agora observando o boxplot dos anos, vemos que apresenta a mesma tendência crescente que vimos no gráfico anterior.

```
data |>
  plot_seasonal_diagnostics(
    .date_var = date,
    .value = chic,
    .interactive = FALSE,
    .title = "")
```



Meses

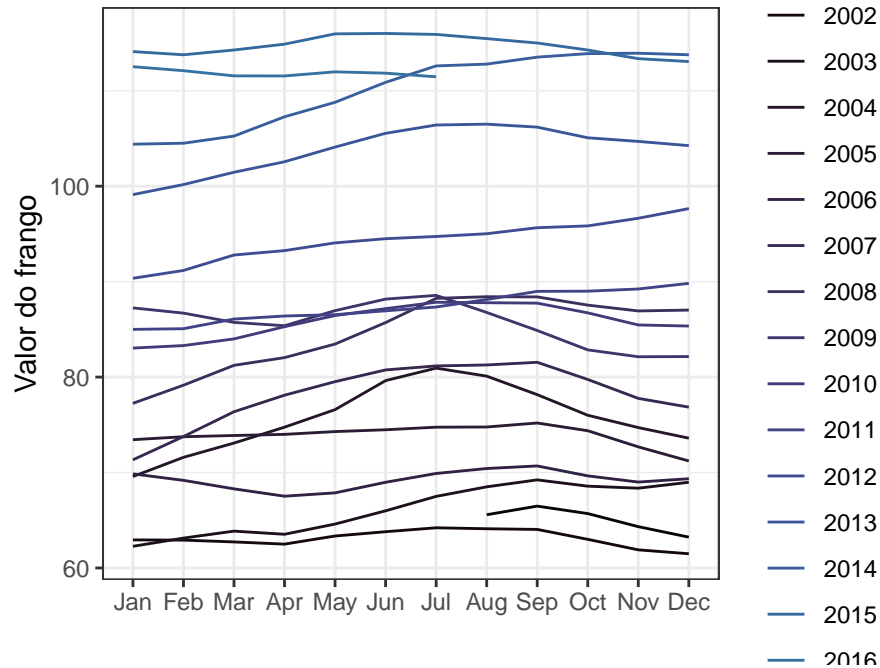
A queda abrupta no ano de 2006 pode ser explicada pela gripe de H5N1 (gripe aviária) que chegou aos EUA quando descobriram patos infectados.

```
data |>
  ggplot(
    aes(
      x = month(date, label = TRUE),
      y = chic,
```

```

    group = as.factor(year(date)),
    color = as.factor(year(date))
  )
) +
geom_line() +
theme_bw() +
labs(y = "Valor do frango", x = "") +
scale_color_manual(values = viridis::mako(33))

```



Explore e descreva todas as informações/características relevantes sobre a série em questão.

- (a) Apresente e analise os gráficos da série e os teste de hipótese que sejam relevantes para descrever a série. Vale ressaltar aqui que todos testes de hipóteses que você considerar, é preciso descrever as hipóteses nula e alternativa, nível de significância adotado e a conclusão do teste.

Teste para sazonalidade

Para confirmar a sazonalidade, ao nível de 5% de significância, foram realizados 3 testes: O teste de Kruskal-Wallis, Friedman e o combinado. Os três testes confirmam a hipótese de sazonalidade que anteriormente foi pensada analisando somente o gráfico da série. Dessa forma, a série é sazonal.

O KW tem como hipótese nula a não sazonalidade. Da mesma forma, o teste de Friedman tem como hipótese nula a não sazonalidade da série.

```
# teste kw
data |>
  pull(chic) |>
  isSeasonal(
    freq = 12,
    test = "kw"
  )
```

[1] TRUE

```
# teste de fried
data |>
  pull(chic) |>
  isSeasonal(
    freq = 12,
    test = "fried"
  )
```

[1] TRUE

```
# teste combinado
data |>
  pull(chic) |>
  isSeasonal(
    freq = 12,
  )
```

[1] TRUE

Teste para tendência

Para confirmar a não estacionariedade da série, ao nível de 5% de significância, foram realizados três testes: ADF, KPSS e o teste de Phillips-Perron. Os três testes confirmam a suposição de não estacionariedade da série. Dessa forma, a série é não estacionária.

A hipótese alternativa do ADF é onde rejeitamos a não estacionariedade da série. Já o KPSS tem hipóteses contrárias ao ADF. O PP teste tem as mesmas hipóteses do ADF.

```
# teste adf
data |>
  pull(chic) |>
  adf.test()
```

Augmented Dickey-Fuller Test

```
data: pull(data, chic)
Dickey-Fuller = -2.3955, Lag order = 5, p-value = 0.4109
alternative hypothesis: stationary
```

```
# teste kpss
data|>
  pull(chic) |>
  kpss.test()
```

Warning in `kpss.test(pull(data, chic))`: p-value smaller than printed p-value

KPSS Test for Level Stationarity

```
data: pull(data, chic)
KPSS Level = 3.3898, Truncation lag parameter = 4, p-value = 0.01
```

```
## Teste pp
pp.test(chicken)
```

Phillips-Perron Unit Root Test

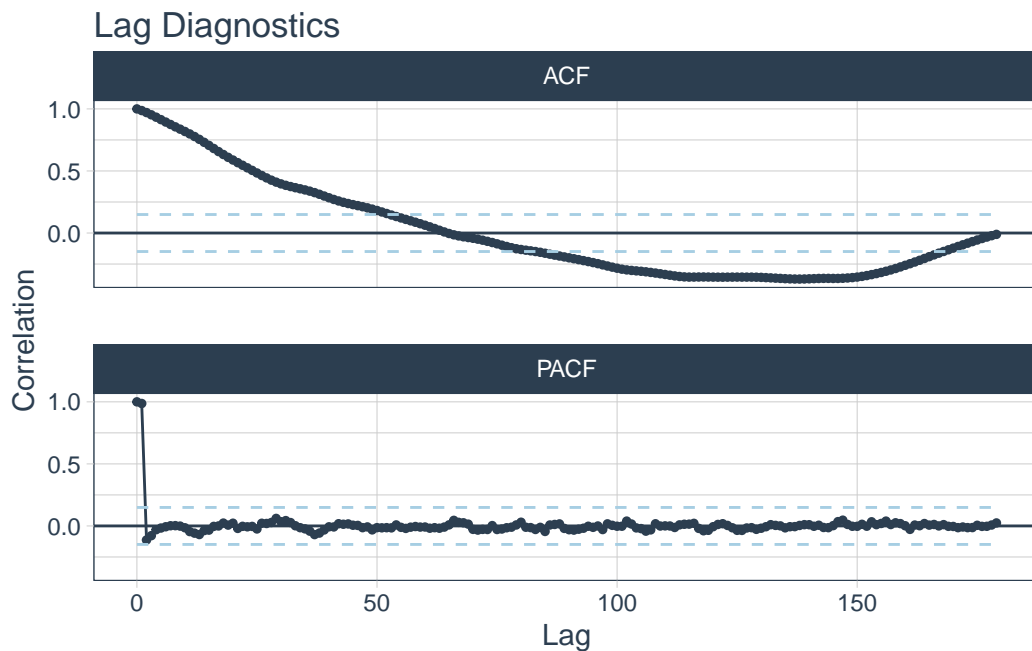
```
data: chicken
Dickey-Fuller Z(alpha) = -11.809, Truncation lag parameter = 4, p-value
= 0.4398
alternative hypothesis: stationary
```

Gráfico de ACF e PACF

Pelo gráfico de gráfico de ACF podemos ver claramente a tendência contida na série que está sendo analisada. No entanto, a sazonalidade é bastante tímida, aparecendo apenas alguns lags com uma correlação um pouco maior que outros.

```
data |>
  plot_acf_diagnostics(.date_var = date, .value = chic, .interactive = FALSE)
```

Max lag exceeds data available. Using max lag: 179



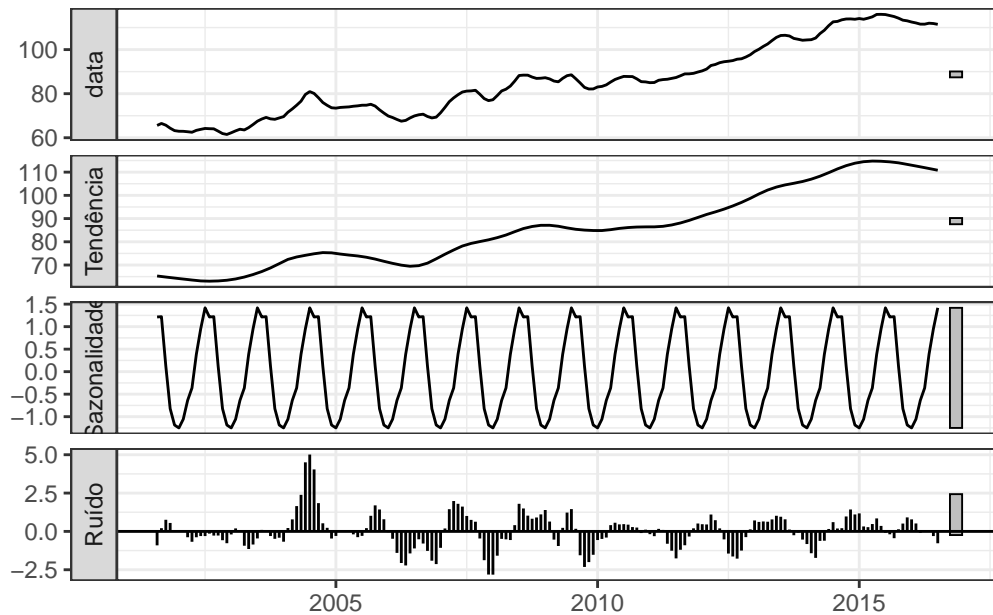
Decomposição

STL

Pela decomposição STL, podemos ver que a maior parte da variação presente na série vem da tendência que está contida nela. Além disso, podemos ver também o que o corre com a sazonalidade, sempre diminuindo o valor das aves aos finais do ano e tendo um padrão de crescimento aos primeiros meses. É possível ver também que a maior alta dos resíduos foi em 2004. Podemos ver também a queda nos resíduos no ano de 2006, possivelmente ocasionado pelo H5N1.

```
STL <- stl(chicken, s.window = "periodic", robust = TRUE)
```

```
STL |>  
  autoplot(  
    labels = c("Tendência", "Sazonalidade", "Ruído")  
  ) +  
  xlab("") +  
  theme_bw()
```



Utilize a função `auto.arima` do pacote `forecast` para selecionar os ordens dos modelos ARIMA/SARIMA.

Estimando os parâmetros pelo AIC e BIC

O modelo obtido pelo critério AIC foi um `Sarima (2, 1, 1)x(0, 0, 1)[12]` with drift, tendo um AIC de 351.01, AICc de 351.5 e BIC 370.14. Pelo critério BIC foi o um `Sarima (2, 1, 0)x(0, 0, 1)[12]`, sem o parâmetro não sazonal MA. Ele teve um AIC de 351.64, AICc de 351.87 e BIC de 364.39. Podemos ver pelas métricas que o modelo obtido pelo BIC aparenta ser melhor que aquele obtido pelo AIC. Pelo critério AICc, o modelo foi o mesmo obtido pelo AIC, por isso foi deixado de fora

```
# AIC
selecao1 = auto.arima(chicken, ic = "aic")

# BIC
selecao2 = auto.arima(chicken, ic = "bic")
```

Verificando a significância dos parâmetros obtidos pelos dois critérios

Agora observando a significância dos parâmetros dos modelos obtidos, vamos que dois parâmetros do modelo obtido pelo AIC, é não significante, o parâmetro de média móveis e o drift, que é uma tendência de longo prazo ou uma média de decaimento presente na série temporal que não é capturado pelo autoregressivo ou média móveis.

Já para o segundo modelo obtido pelo BIC, vemos que ao nível de 5% de significância, todos os parâmetros são significantes para o modelo, até mesmo a componente de média móvel sazonal. Assim, o modelo selecionado será aquele pelo método BIC.

```
# modelo 1
coeftest(selecao1)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
ar1	1.293318	0.222046	5.8245	5.727e-09	***
ar2	-0.537455	0.154224	-3.4849	0.0004923	***
ma1	-0.401941	0.256885	-1.5647	0.1176595	
sma1	0.275629	0.069237	3.9809	6.864e-05	***
drift	0.251840	0.142755	1.7641	0.0777073	.

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# modelo 2
coeftest(selecao2)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)
ar1	0.930160	0.073223	12.7031	< 2.2e-16 ***
ar2	-0.258929	0.073074	-3.5434	0.000395 ***
sma1	0.287094	0.067203	4.2721	1.937e-05 ***

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cheque a qualidade do ajuste do(s) modelo(s) selecionado(s) no item anterior e explique cada os resultados obtidos para avaliar a qualidade do ajuste.

Qual a conclusão do teste de Ljung e Box (apresente as hipótese)? Comente.

Ao nível de 5% de significância, vemos que os resíduos são independentes pois não rejeitam a hipótese nula de independência dos resíduos. Isso acontece para o teste de Ljung-Box e para o Box-Pierce, que tem as mesmas hipóteses.

```
# Ljung-Box
Box.test(selecao2$residuals, type = "Ljung-Box")
```

Box-Ljung test

```
data: selecao2$residuals
X-squared = 0.43018, df = 1, p-value = 0.5119
```

```
# Box-Pierce
Box.test(selecao2$residuals, type = "Box-Pierce")
```

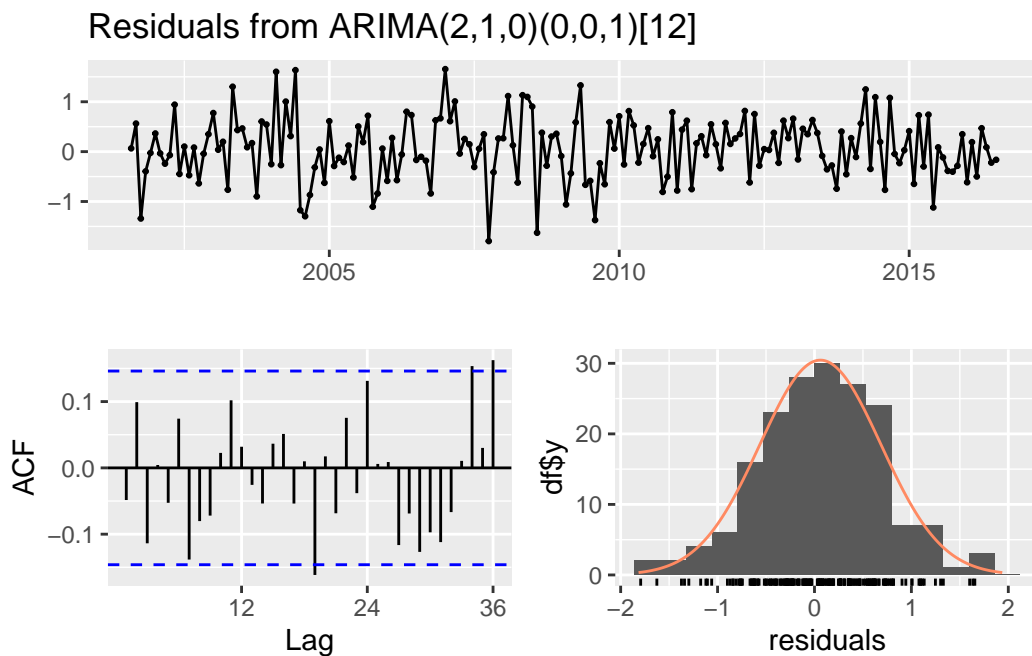
Box-Pierce test

```
data: selecao2$residuals  
X-squared = 0.42309, df = 1, p-value = 0.5154
```

Apresente os gráficos do diagnóstico. Interprete e comente.

Observando os ACF dos resíduos, vemos que eles estão entre os limites, mostrando que os resíduos de fato são ruídos branco. Pelo histograma dos resíduos, vemos que de fato eles aparentam seguir a distribuição normal. Além disso, os resíduos observando o gráfico de linha, de fato aparentam ser não independentes. No entanto, há um lag que passa do limite inferior no ACF, talvez tendo sido ocasionado por algum outlier da série temporal, podendo ser do ano de 2006.

```
checkresiduals(selecao2)
```



Ljung-Box test

```
data: Residuals from ARIMA(2,1,0)(0,0,1)[12]
```

$Q^* = 27.878$, $df = 21$, $p\text{-value} = 0.1436$

Model df: 3. Total lags used: 24

Pelo gráfico do p-valor do teste de Ljung-Box, vemos que de fato os resíduos são independentes, pois não ultrapassam o limite inferior que está no gráfico.

Gráfico criado com o tsdiag é muito grande e não cabe no PDF, por isso foi omitido, apesar de ter comentários sobre ele

```
# Gráfico criado com o tsdiag é muito grande e não cabe no PDF, por isso foi omitido, apesar de ter comentários sobre ele  
  
# tsdiag(selecao2)
```

Exclua as 12 últimas observações da série temporal e faça previsão considerando a série truncada (chicken.truc).

(a) Considere a série (chicken.truc) e apresente os valores previstos 12 passos a frente considerando os modelos ARIMA/SARIMA.

Importante: Note que é preciso buscar novos ajustes/checar qualidade do ajuste para chicken.truc, e depois fazer as previsões.

Arima

```
# 12 últimas obs. (esse é a série para o teste, os 12 passos previstos)  
chicken.truc <- data |>  
  tail(12)  
# restante da série para fazer o ajuste do modelo ou métodos (aqui é o banco de treinamento)  
chicken.train = data[1:168,]  
  
# modelo criado com modeltime:  
model_spec_arima <- arima_reg(  
  non_seasonal_ar = 2,  
  non_seasonal_differences = 1,  
  non_seasonal_ma = 0,  
  seasonal_ar = 0,  
  seasonal_differences = 0,  
  seasonal_ma = 1) |>
```

```

set_engine("arima") |>
fit(log(chic) ~ date, data = chicken.train) |>
modeltime_table() |>
modeltime_calibrate(new_data = chicken.truc)

```

frequency = 12 observations per 1 year

```

model_spec_arima |>
modeltime_forecast(
  new_data = chicken.truc,
  actual_data = data
) %>%
plot_modeltime_forecast(
  .interactive = FALSE
)

```

Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -Inf

Forecast Plot



Considere a série (chicken.truc) e apresente os valores previstos 12 passos a frente considerando o algoritmo de alisamento exponencial.

Holt-Winters

O Alisamento exponencial a ser utilizado será o de Holt-Winters, pois ele conseguirá representar a tendência e sazonalidade presente na série temporal. Será considerado o caso multiplicativo.

```
model_fit_ets <- exp_smoothing(  
  seasonal_period = 12,  
  season = "multiplicative",  
  trend = "multiplicative",  
  error = "auto",  
) |>  
  set_engine("ets") |>  
  fit(chic ~ date, data = chicken.train)  
model_tbl_ets <- modeltime_table(model_fit_ets)  
  
model_tbl_ets |>  
  modeltime_calibrate(new_data = chicken.truc) |>  
  modeltime_forecast(  
    new_data = chicken.truc,  
    actual_data = data  
  ) |>  
  plot_modeltime_forecast(.interactive = FALSE, .title = "")
```

Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -Inf



```
ets_metrics <- model_tbl_ets |>
  modeltime_calibrate(new_data = chicken.truc) |> modeltime_accuracy()
```

(c) De acordo com as características da série, utilize os métodos simples de previsão considerando a série `chicken.truc` e apresente os valores previstos 12 passos a frente.

O método simples escolhido foi o naive sazonal e o média móveis simples.

Naive sazonal

```
model_fit_snaive <- naive_reg(seasonal_period = 12) |>
  set_engine("snaive") |>
  fit(chic ~ date, data = chicken.train)
model_tbl_snaive <- modeltime_table(model_fit_snaive)

model_tbl_snaive |>
  modeltime_calibrate(new_data = chicken.truc) |>
  modeltime_forecast(
```

```

    new_data = chicken.truc,
    actual_data = data
  ) |>
  plot_modeltime_forecast(.interactive = FALSE, .title = "")

```

Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -Inf



```

snaive_metrics <- model_tbl_snaive |>
  modeltime_calibrate(new_data = chicken.truc) |> modeltime_accuracy()

```

Média móveis simples

```

n <- nrow(data)
media_movel <- chicken.truc
media_movel$chic <- smooth::sma(chicken.truc$chic, order = 12, h = 12)$forecast

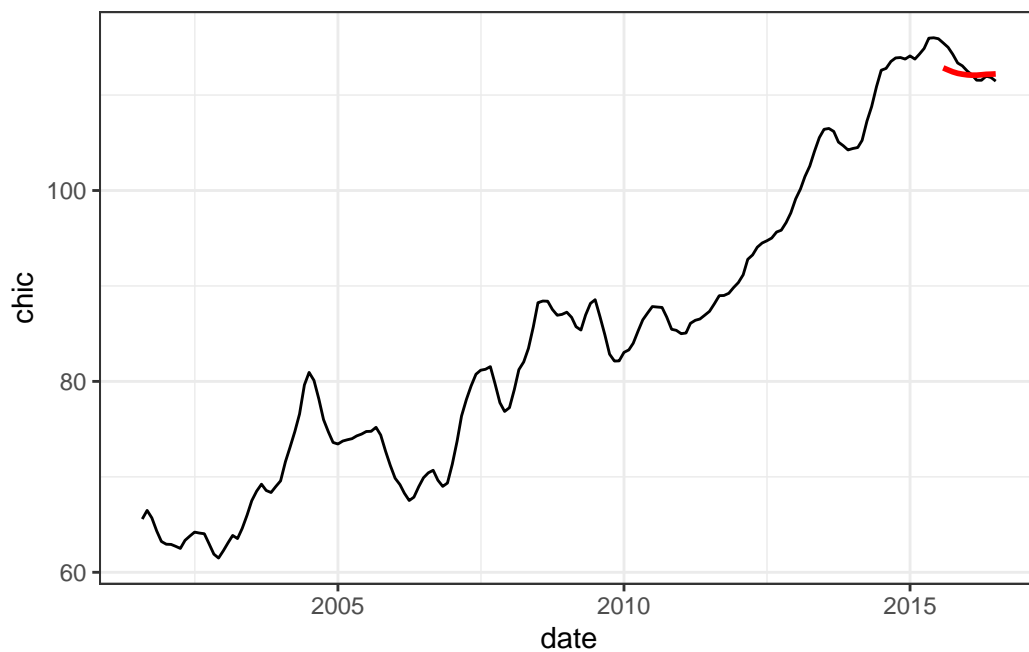
```

Registered S3 method overwritten by 'greybox':


```
method      from  
print.pcor lava
```

```
data |>  
  ggplot(aes(x = date, y = chic)) +  
  geom_line() +  
  geom_line(  
    data = media_movel,  
    aes(x = date, y = chic),  
    color = "red", size = 1,  
  ) +  
  theme_bw()
```

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.



```
rmse_ma <- rmse_vec(chicken.truc$chic, media_movel$chic)  
rsq_ma <- rsq_vec(chicken.truc$chic, media_movel$chic)  
mape_ma <- mape_vec(chicken.truc$chic, media_movel$chic)
```

Observando os dois gráficos anteriores, podemos ver que o método simples que melhor se ajustou a série temporal foi o do naive sazonal. Já a média móvel simples ficou apenas uma reta variado um pouco. Isso pode nos indicar que a componente sazonal da série é aquela que mais influencia no valor das aves.

(d) Calcule e compare os erro quadrático médio e o erro médio percentual (ver função `accuracy`) das previsões obtidas via modelos ARIMA, alisamento exponencial e métodos simples de previsão (utilizados no item anterior). O que você pode concluir?

```
# MÉTRICAS DO MODELO E MÉTODOS ESCOLHIDOS

# arima
metricas_arima <- model_spec_arima |>
  modeltime_accuracy()

# alisamento exponencial
ets_metrics <- model_tbl_ets |>
  modeltime_calibrate(new_data = chicken.truc) |> modeltime_accuracy()

# naive sazonal

snaive_metrics <- model_tbl_snaive |>
  modeltime_calibrate(new_data = chicken.truc) |> modeltime_accuracy()

# metricas MAS
rmse_ma <- rmse_vec(chicken.truc$chic, media_movel$chic)
rsq_ma <- rsq_vec(chicken.truc$chic, media_movel$chic)
mape_ma <- mape_vec(chicken.truc$chic, media_movel$chic)
```

Os melhores resultados foram obtidos do modelo Arima e média móvel simples. No entanto, o Arima possui uma variância menor que a média móvel simples e tem um MAPE também menor que o média móvel simples. O último melhor foi o de Holt-Winters, mas ainda obteve resultados mais confiáveis que o naive sazonal, pois tem um MAPE e RMSE menor.

Métricas do modelo:

- **Arima**
- R2: 76%

- RMSE: 0.0298
- MAPE: 0.566
- **Holt-Winters**
- R2: 12.7%
- RMSE: 1.78
- MAPE: 1.25
- **Naive sazonal**
- R2: 56.1%
- MAPE: 2.05
- RMSE: 2.69
- **Médias móveis simples**
- R2: 83.7%
- MAPE: 0.8471
- RMSE: 1.26