ARIMA Models

Lecture 09

Dr. Colin Rundel

MA(∞)

MA(q)

From last time - a $\text{MA}(q)$ process with $w_t \overset{\text{iid}}{\sim} N(0, \sigma_w^2)$,

$$y_t = \delta + w_t + \theta_1 \, w_{t-1} + \theta_2 \, w_{t-2} + \cdots + \theta_q \, w_{t-q}$$

has the following properties,

$$E(y_t) = \delta$$

$$Var(y_t) = \gamma(0) = (1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2) \, \sigma_w^2$$

$$Cov(y_t, y_{t+h}) = \gamma(h) = \begin{cases} \sigma_w^2 \sum_{j=0}^{q-|h|} \theta_j \theta_{j+|h|} & \text{if } |h| \le q \\ 0 & \text{if } |h| > q \end{cases}$$

and is stationary for any values of $(\theta_1, \ldots, \theta_q)$

MA(∞)

If we let $q \to \infty$ then process will be stationary if and only if the moving average coefficients (θ 's) are square summable, i.e.

$$\sum_{i=1}^{\infty} \theta_i^2 < \infty$$

which is necessary so that the $Var(y_t) < \infty$ condition is met for weak stationarity.

Sometimes, a slightly stronger condition known as absolute summability, $\sum_{i=1}^{\infty} |\theta_i| < \infty$ is necessary (e.g. for some CLT related asymptotic results).

Invertibility

If an $\text{MA}(q)$ process, $y_t = \delta + \theta_q(L) w_t$, can be rewritten as a stationary AR process then the process is said to be invertible.

$\text{MA}(1)$ w/ $\delta = 0$ example:

Invertibility vs Stationarity

A $\text{MA}(q)$ process is *invertible* if $y_t = \delta + \theta_q(L)\, w_t$ can be rewritten as an exclusively AR process (of possibly infinite order), i.e. $\phi(L)\, y_t = \alpha + w_t$.

Conversely, an $\text{AR}(p)$ process is *stationary* if $\phi_p(L)\, y_t = \delta + w_t$ can be rewritten as an exclusively MA process (of possibly infinite order), i.e. $y_t = \delta + \theta(L)\, w_t$.

So using our results w.r.t. $\phi(L)$ it follows that if all of the roots of $\theta_q(L)$ are outside the complex unit circle then the moving average process is invertible.

Differencing

Difference operator

We will need to define one more notational tool for indicating differencing

$$\Delta y_t = y_t - y_{t-1}$$

Just like the lag operator we will indicate repeated applications of this operator using exponents

$$\begin{aligned}
\Delta^2 y_t &= \Delta(\Delta y_t) \\
&= (\Delta y_t) - (\Delta y_{t-1}) \\
&= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\
&= y_t - 2y_{t-1} + y_{t-2}
\end{aligned}$$

Note that Δ can even be expressed in terms of the lag operator L,

$$\Delta^d = (1 - L)^d$$

Differencing and Stocastic Trend

Using the two component time series model

$$y_t = \mu_t + x_t$$

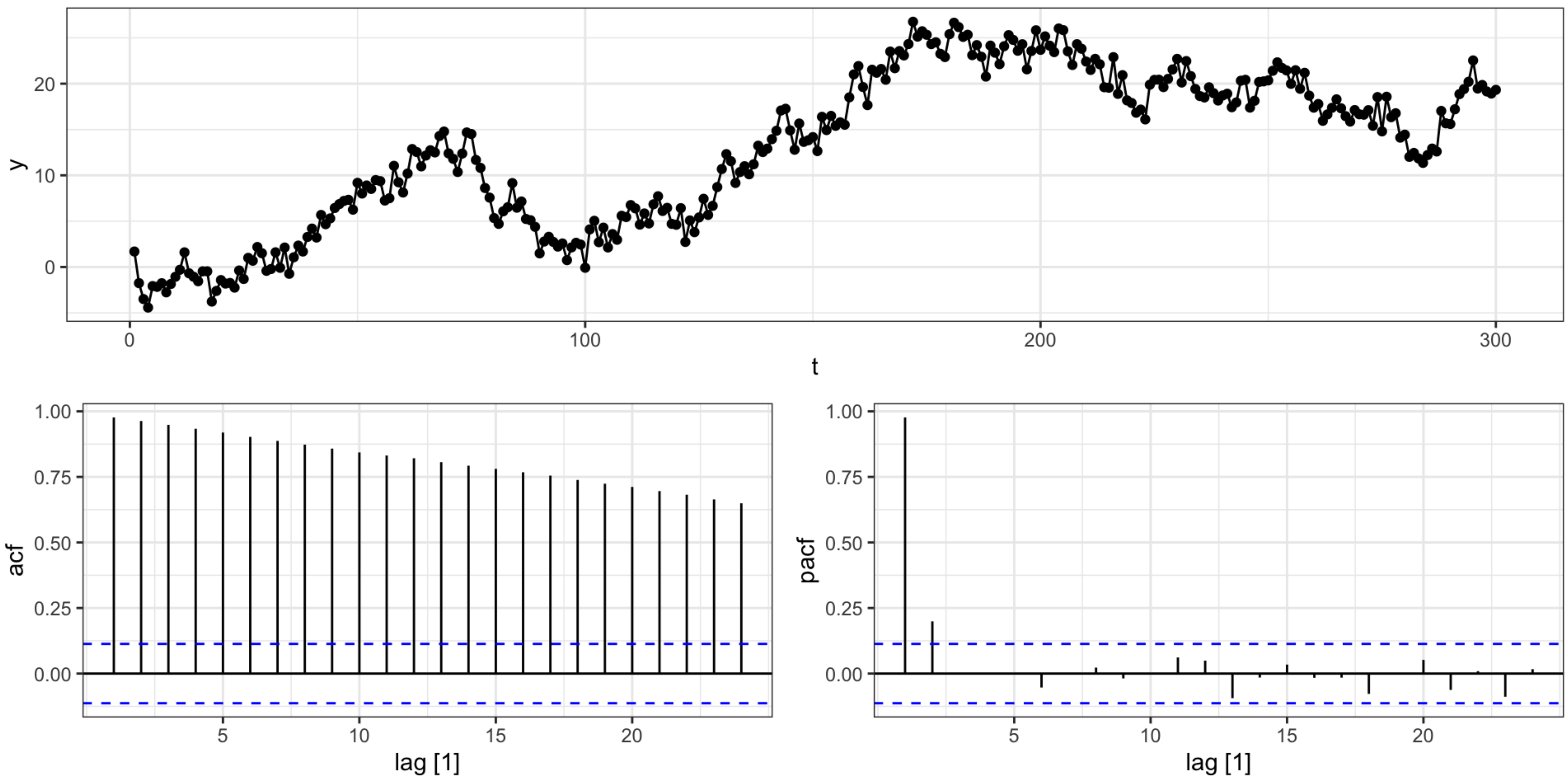
where μ_t is a non-stationary trend component and x_t is a mean zero stationary component.

We have already shown that differencing can address deterministic trend (e.g. $\mu_t = \beta_0 + \beta_1\, t$). In fact, if μ_t is any k-th order polynomial of t then $\Delta^k y_t$ is stationary.

Differencing can also address stochastic trend such as in the case where μ_t follows a random walk.

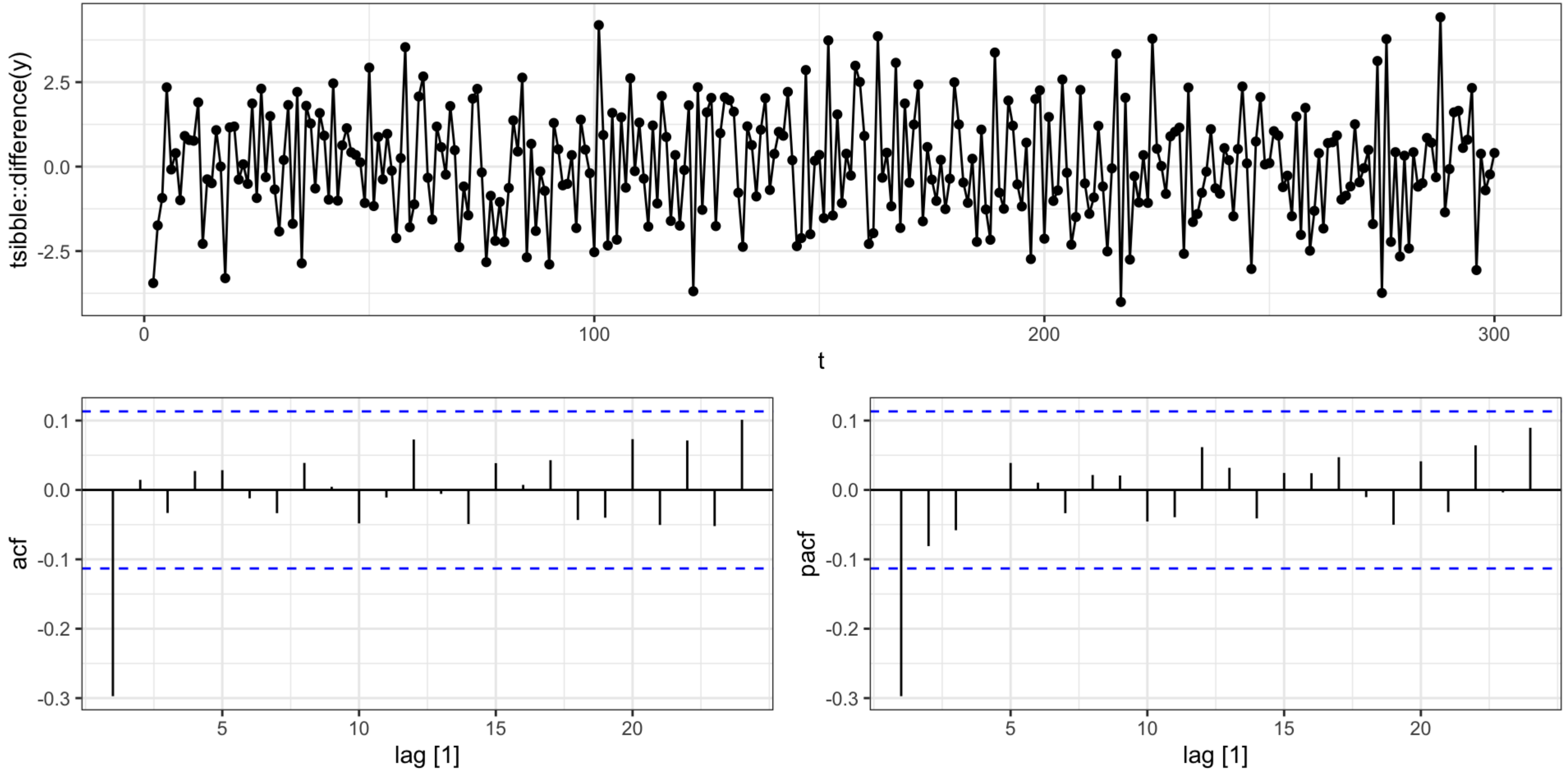
Stochastic trend - Example 1

Let $y_t = \mu_t + w_t$ where w_t is white noise and $\mu_t = \mu_{t-1} + v_t$ with v_t being a stationary process with mean 0.



Differenced stochastic trend

```
feasts::gg_tsdisplay(d, y = tsibble::difference(y), plot_type = "partial")
```



Stationary?

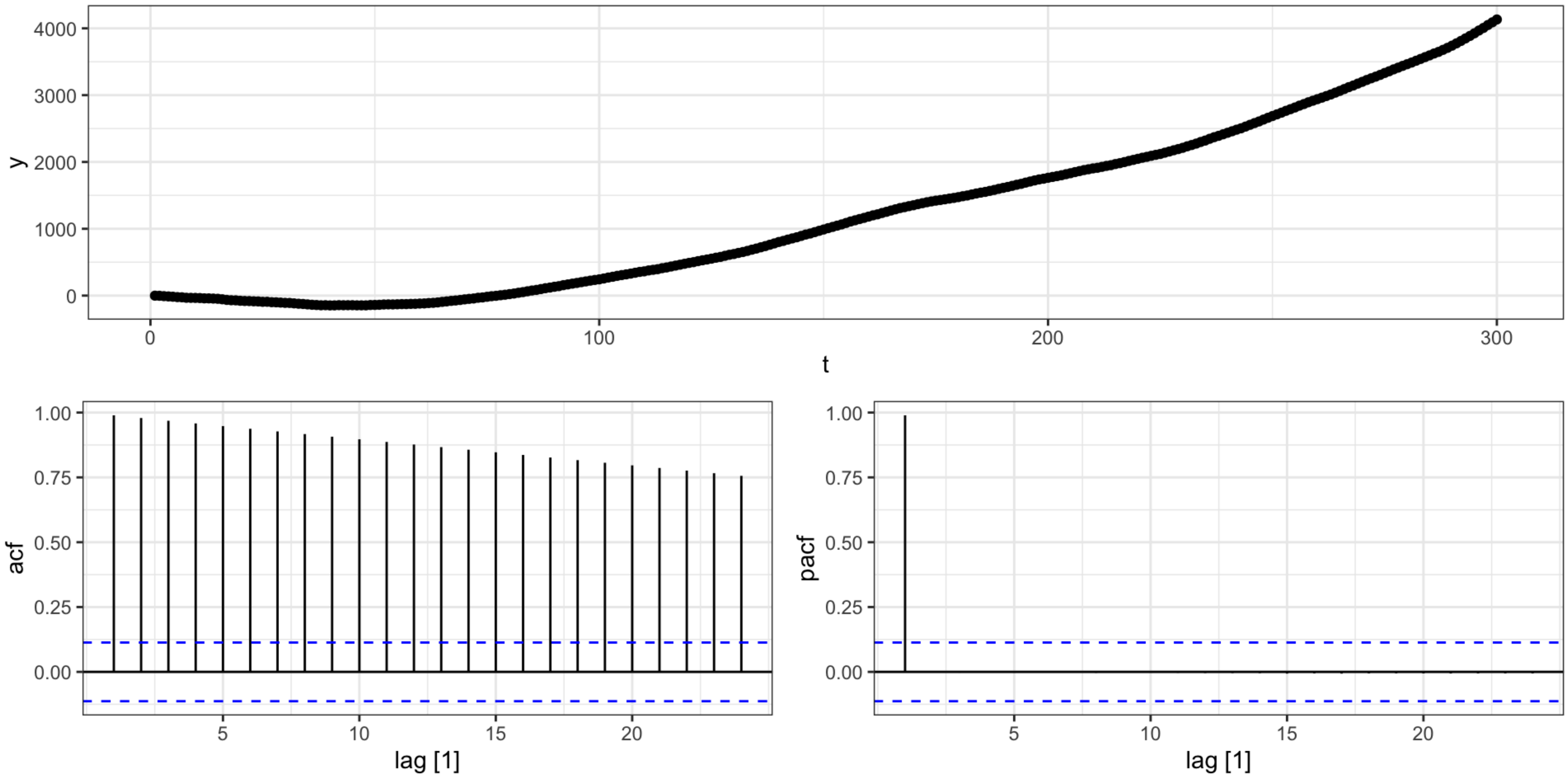
Is y_t stationary?

Difference Stationary?

Is Δy_t stationary?

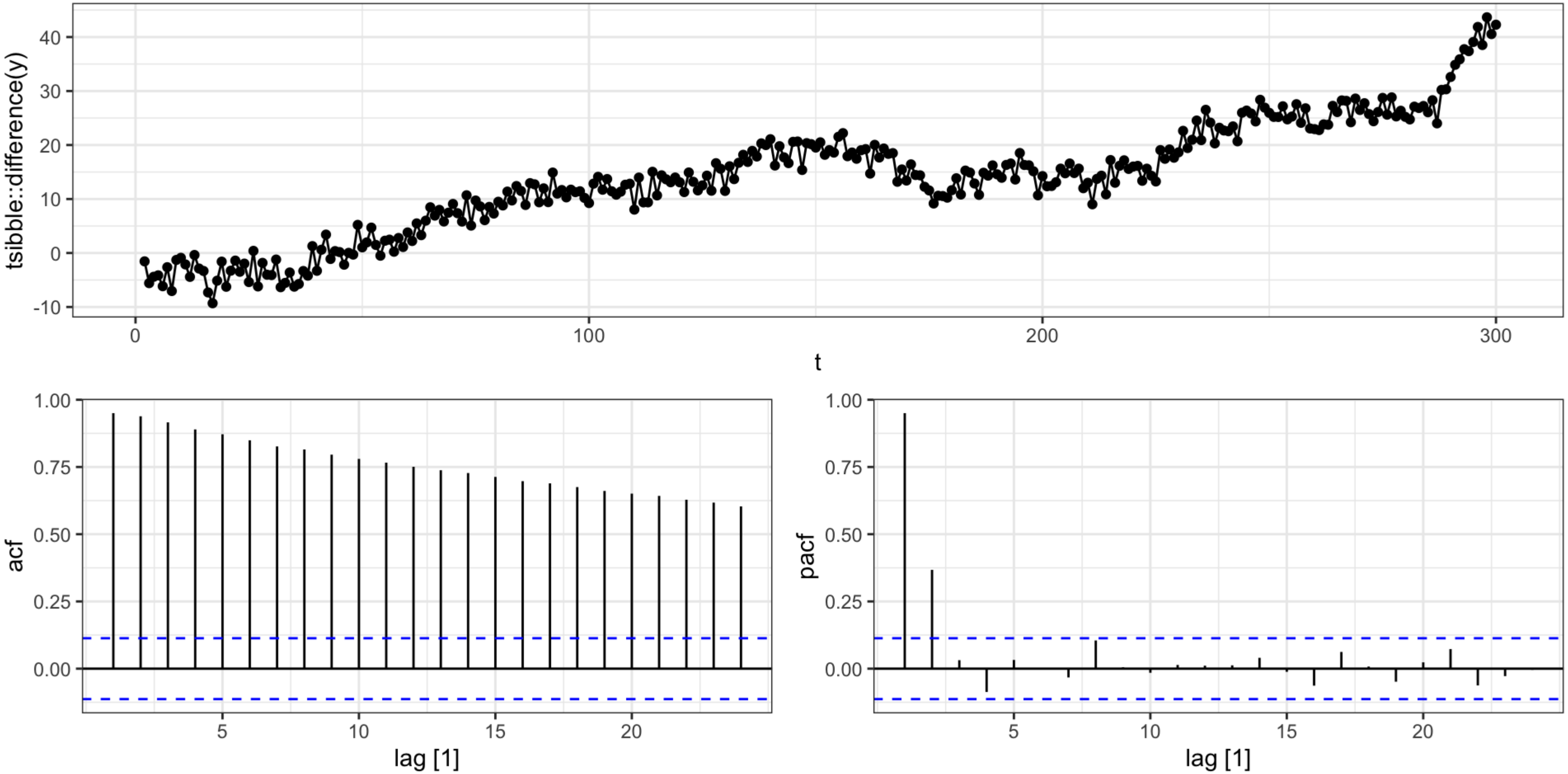
Stochastic trend - Example 2

Let $y_t = \mu_t + w_t$ where w_t is white noise and $\mu_t = \mu_{t-1} + v_t$ but now $v_t = v_{t-1} + e_t$ with e_t being stationary.



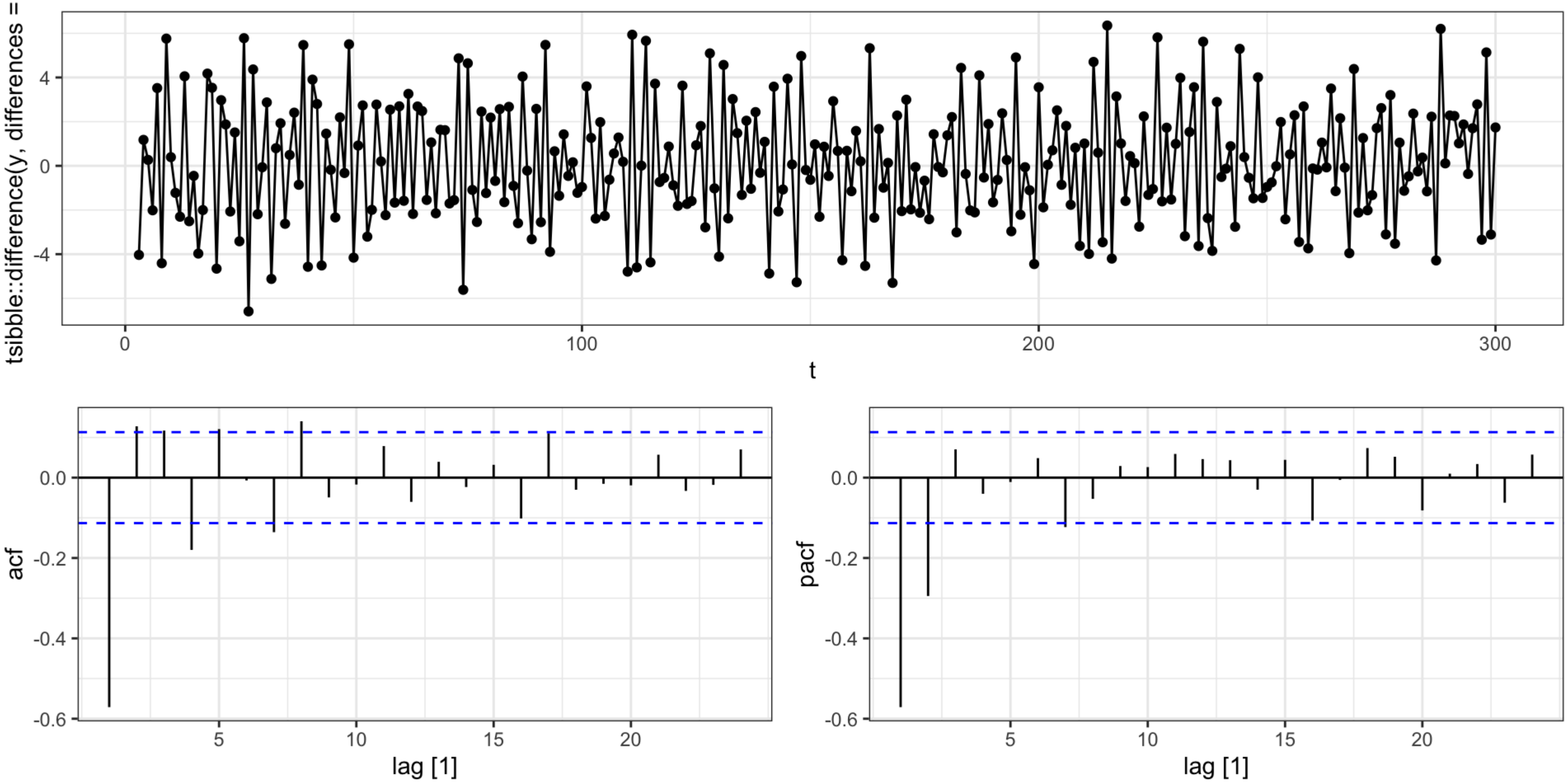
Differenced stochastic trend

```
feasts::gg_tsdisplay(d, y = tsibble::difference(y), plot_type = "partial")
```



Twice differenced stochastic trend

```
feasts::gg_tsdisplay(d, y = tsibble::difference(y, differences = 2), plot_type = "partial")
```



Difference stationary?

Is Δy_t stationary?

2nd order difference stationary?

What about $\Delta^2 y_t$, is it stationary?

ARIMA

ARIMA Models

Autoregressive integrated moving average are just an extension of an ARMA model to include differencing of degree d to y_t before including the autoregressive and moving average components.

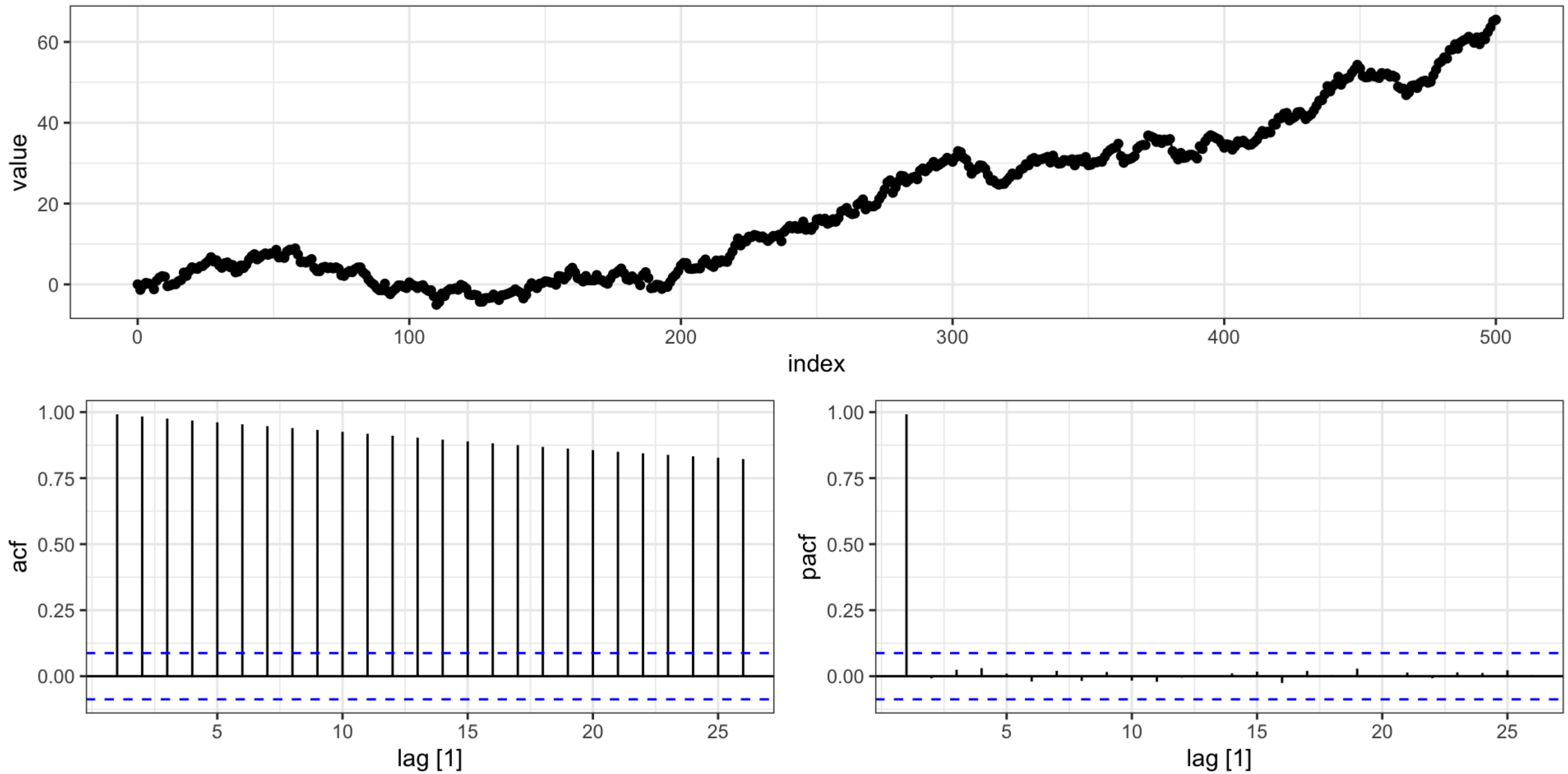
$$\text{ARIMA}(p, d, q): \qquad \phi_p(L)\, \Delta^d\, y_t = \delta + \theta_q(L) w_t$$

Box-Jenkins approach:

1. Transform data if necessary to stabilize variance
2. Choose order (p, d, q) of ARIMA model
3. Estimate model parameters (δ, ϕs, and θs)
4. Diagnostics

Random walk

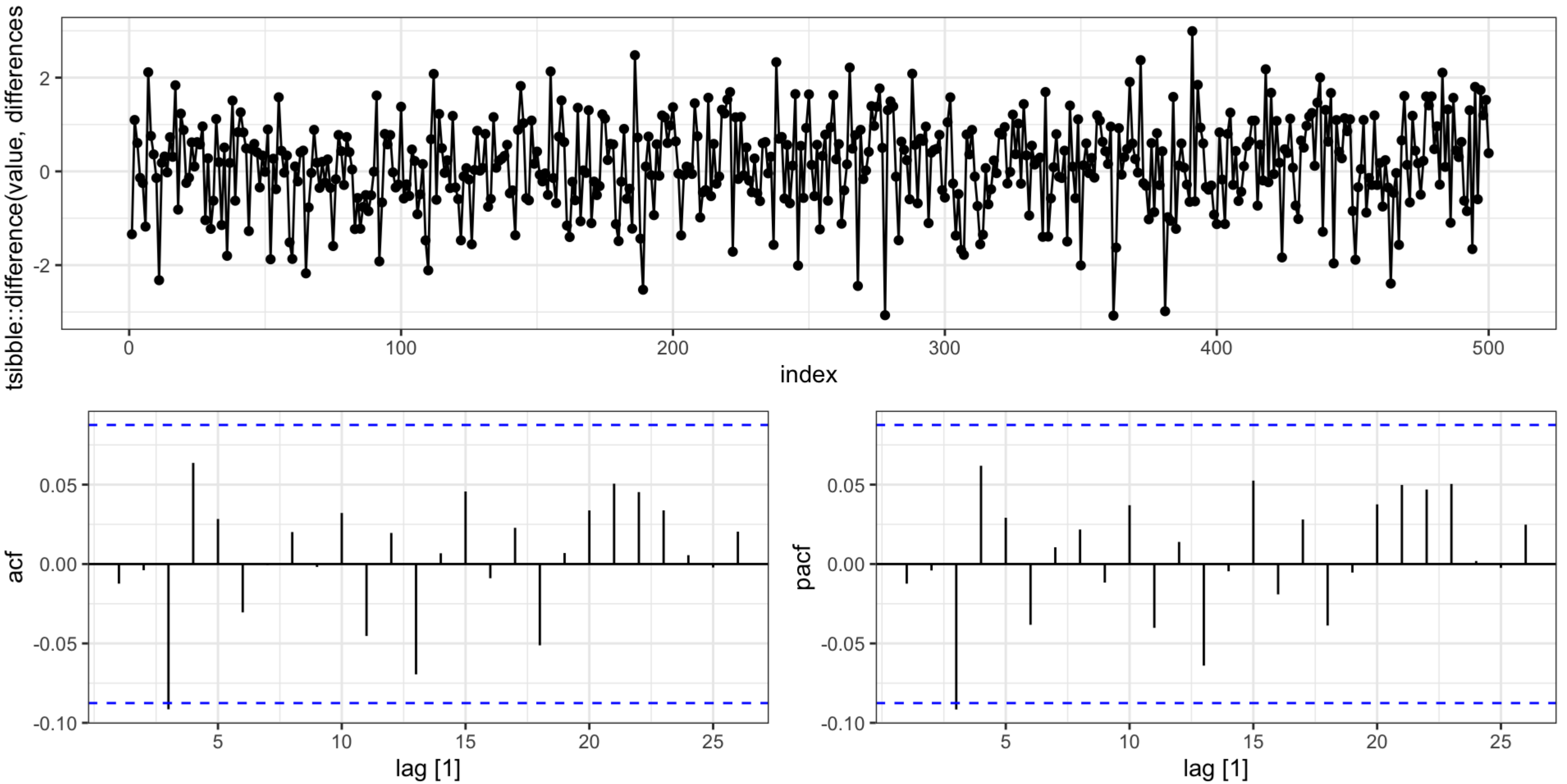
```r
rwd = arima.sim(n=500, model=list(order=c(0,1,0)), mean=0.1) |> tsibble::as_tsibble()
feasts::gg_tsdisplay(rwd, y=value, plot_type = "partial")
```



Differencing

differences = 1 | differences = 2 | differences = 3

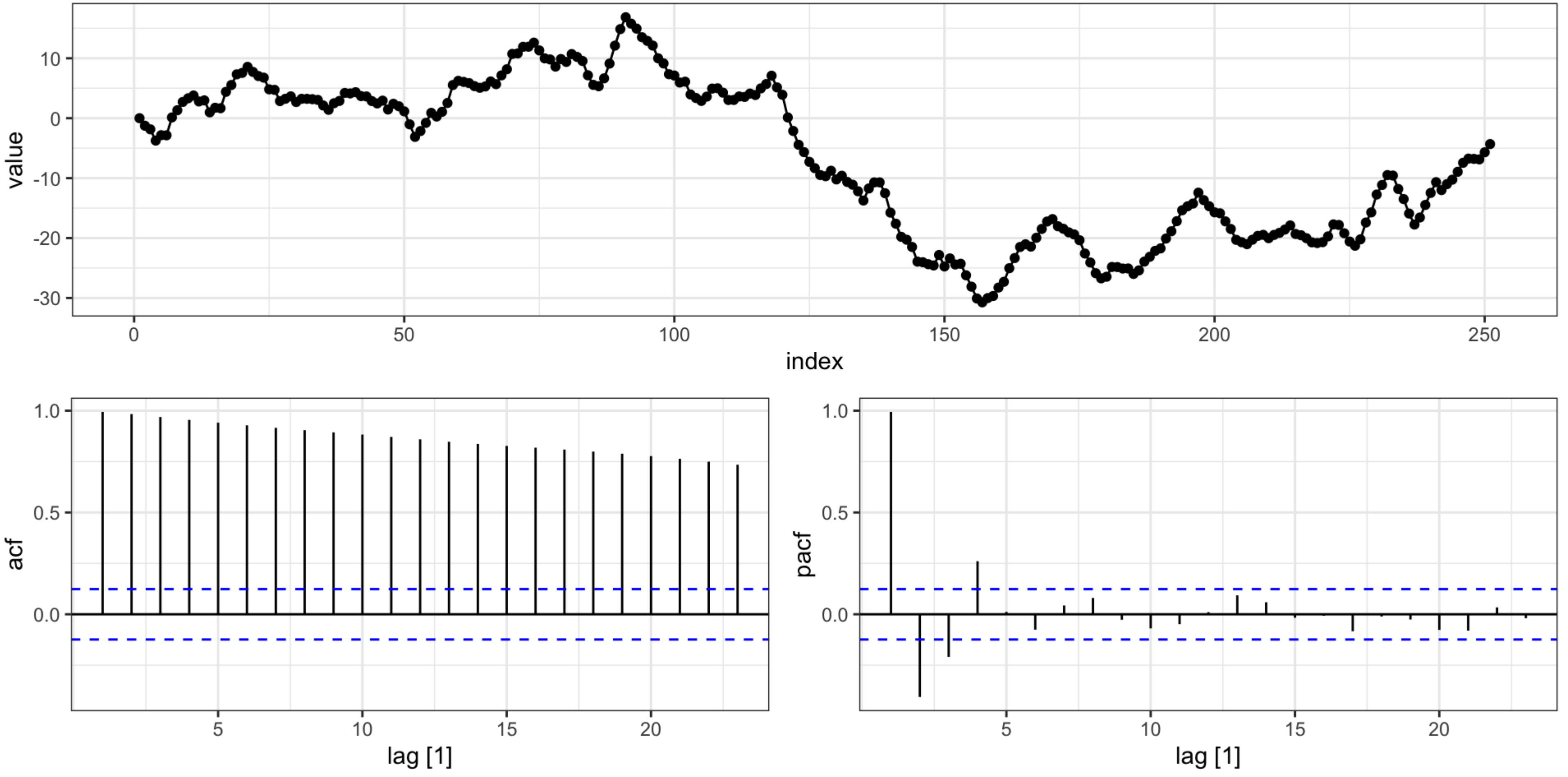
```
feasts::gg_tsdisplay(rwd, y=tsibble::difference(value, differences = 1), plot_type = "partial")
```



AR or MA?

ts1

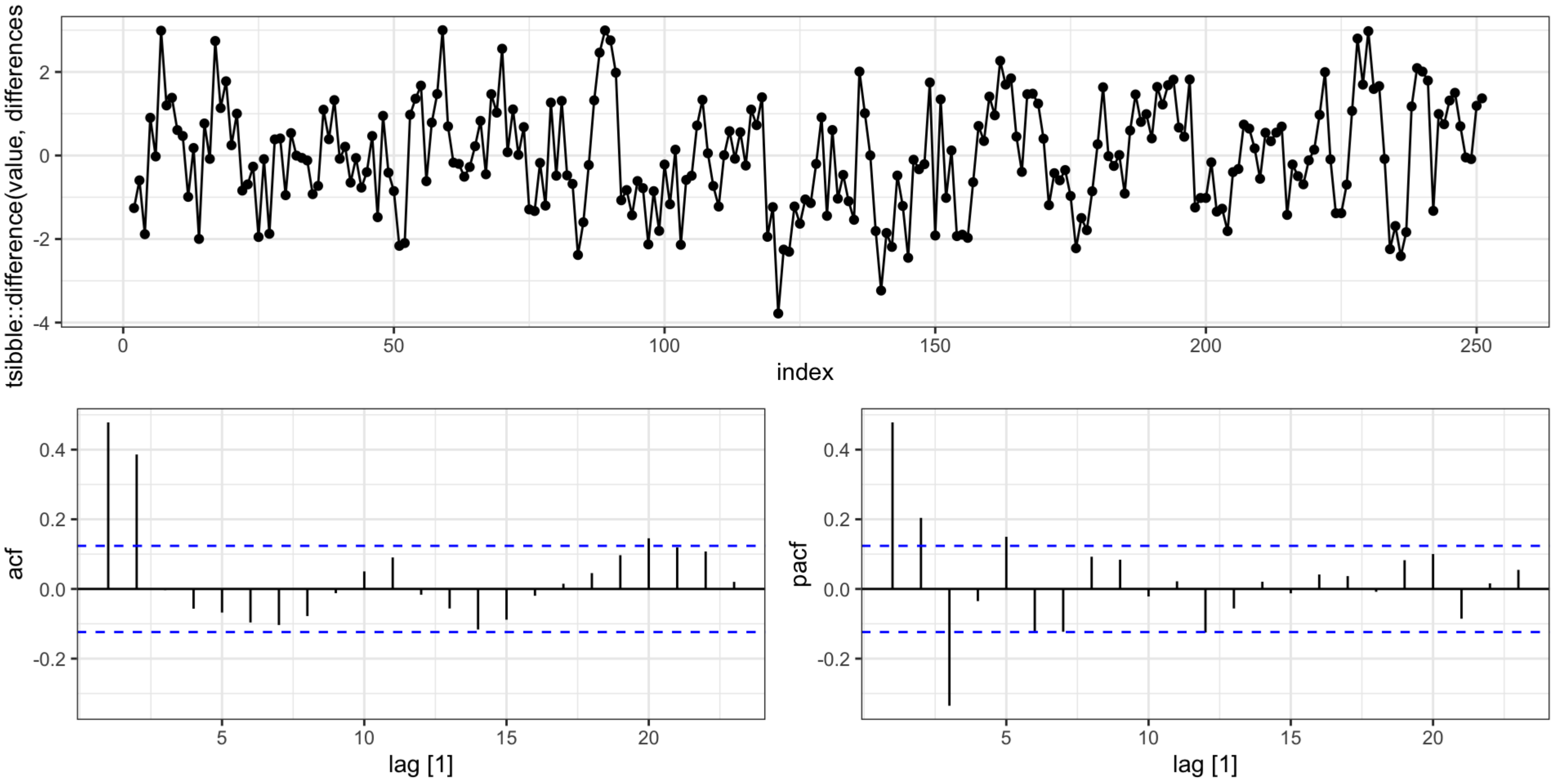
```r
feasts::gg_tsdisplay(ts1, y=value, plot_type = "partial")
```



ts1 - Finding d

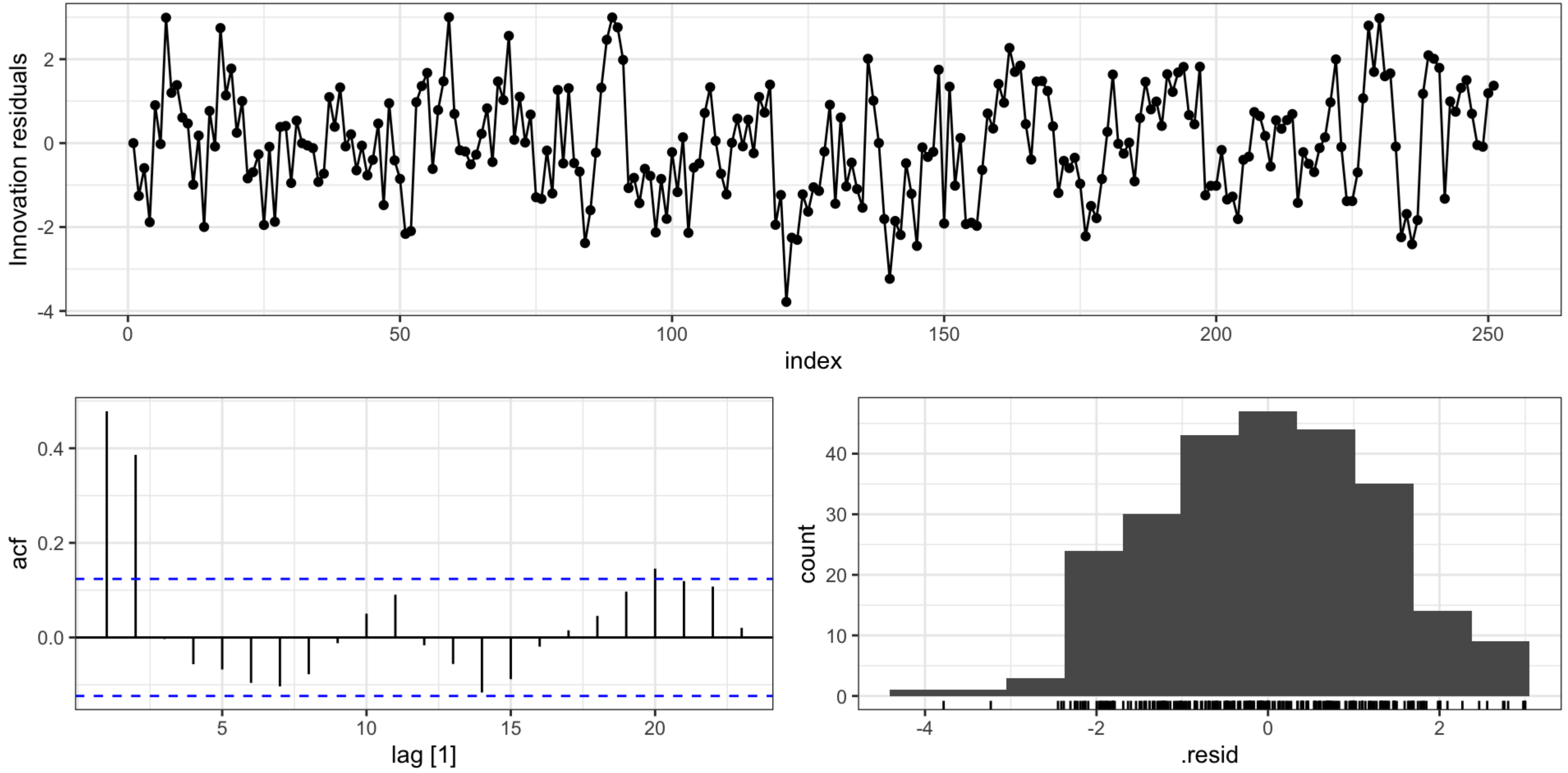
differences = 1 | differences = 2 | differences = 3

```
feasts::gg_tsdisplay(ts1, y=tsibble::difference(value, differences = 1), plot_type = "partial")
```



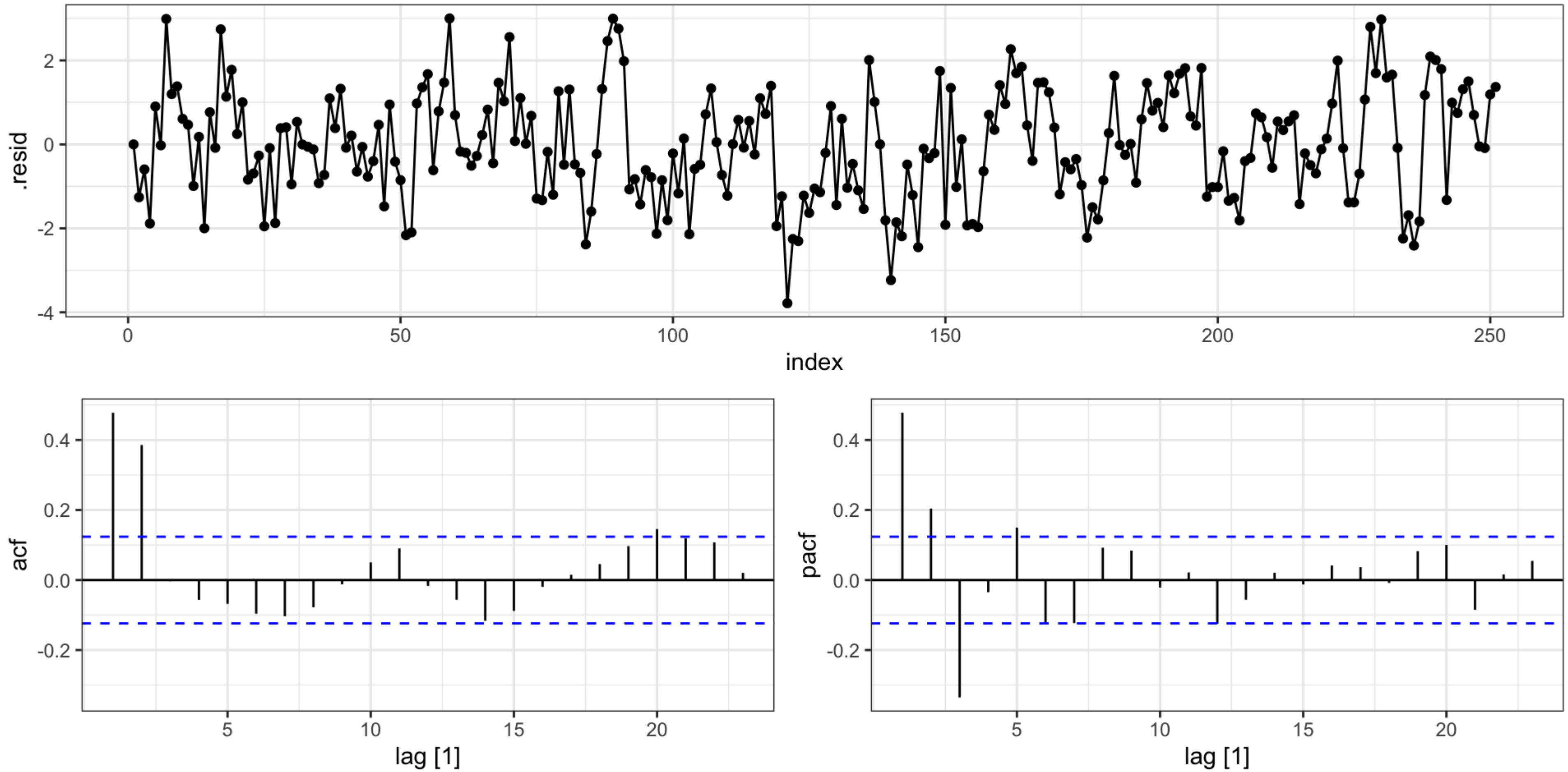
Residuals - `ts1`- `ARIMA(0,1,0)`

```r
model(ts1, ARIMA(value ~ pdq(0,1,0))) |>
  feasts::gg_tsresiduals()
```



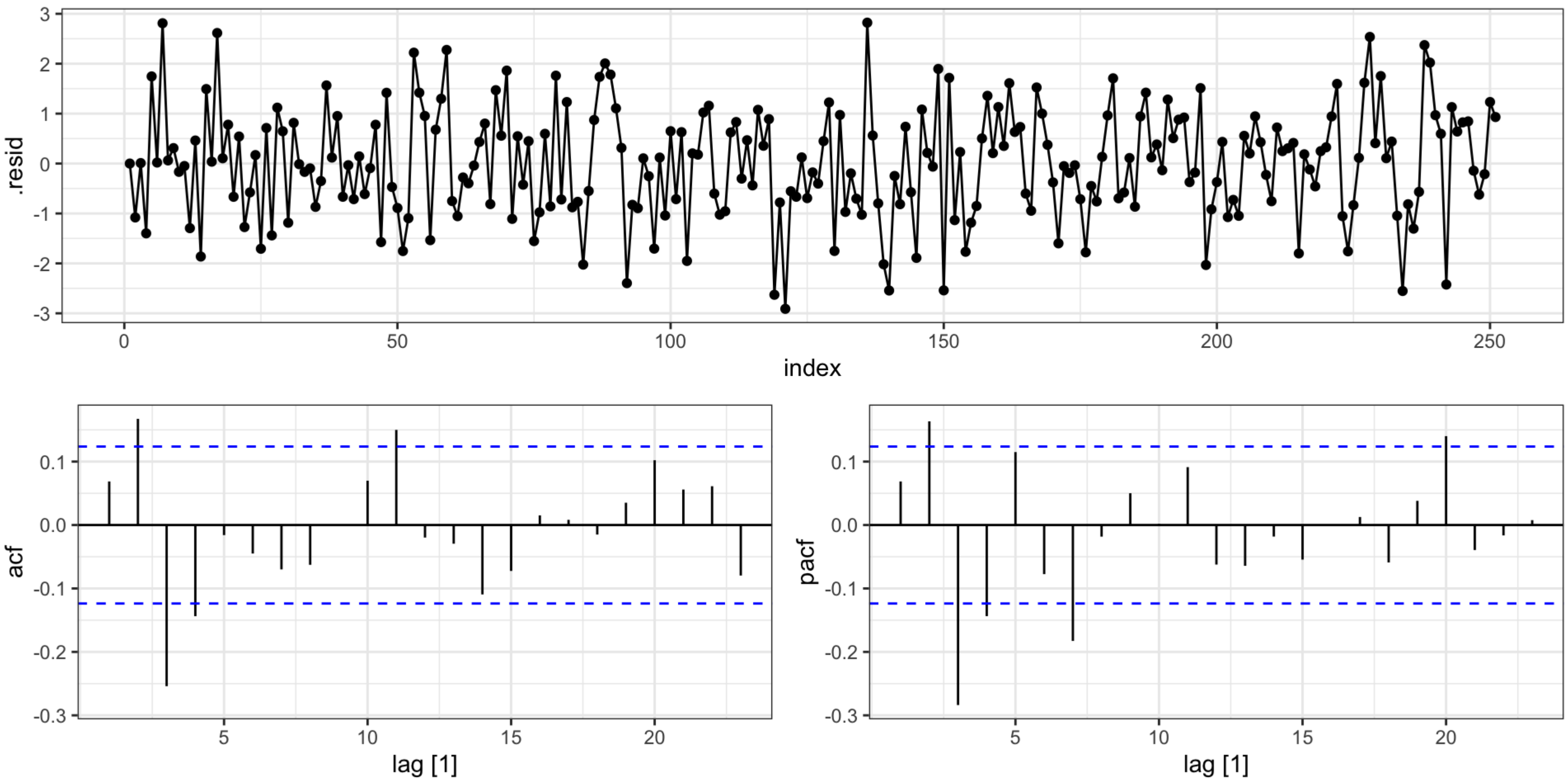
Residuals - `ts1` - `ARIMA(0,1,0)`

```
model(ts1, ARIMA(value ~ pdq(0,1,0))) |>
  residuals() |>
  feasts::gg_tsdisplay(y = .resid, plot_type = "partial")
```



Residuals - `ts1` - `ARIMA(2,1,0)`

```r
model(ts1, final = ARIMA(value ~ pdq(2,1,0))) |>
  residuals() |>
  feasts::gg_tsdisplay(y = .resid, plot_type = "partial")
```



`ts1` - Model comparison

```
model(
  ts1,
  ARIMA(value ~ pdq(0,1,0)), ARIMA(value ~ pdq(1,1,0)), ARIMA(value ~ pdq(0,1,1)),
  ARIMA(value ~ pdq(1,1,1)), ARIMA(value ~ pdq(2,1,0)), ARIMA(value ~ pdq(0,1,2)),
  ARIMA(value ~ pdq(2,1,1)), ARIMA(value ~ pdq(1,1,2)), ARIMA(value ~ pdq(2,1,2))
) |>
  glance()
```

```
# A tibble: 9 × 8
  .model                        sigma2 log_lik   AIC  AICc   BIC ar_roots ma_roots
  <chr>                          <dbl>   <dbl> <dbl> <dbl> <dbl> <list>   <list>
1 ARIMA(value ~ pdq(0, 1, 0))     1.69   -421.  843.  843.  847. <cpl>    <cpl>
2 ARIMA(value ~ pdq(1, 1, 0))     1.31   -388.  780.  780.  787. <cpl>    <cpl>
3 ARIMA(value ~ pdq(0, 1, 1))     1.46   -402.  807.  807.  814. <cpl>    <cpl>
4 ARIMA(value ~ pdq(1, 1, 1))     1.29   -386.  777.  777.  788. <cpl>    <cpl>
5 ARIMA(value ~ pdq(2, 1, 0))     1.26   -382.  771.  771.  782. <cpl>    <cpl>
6 ARIMA(value ~ pdq(0, 1, 2))     1.06   -361.  728.  728.  739. <cpl>    <cpl>
7 ARIMA(value ~ pdq(2, 1, 1))     1.20   -376.  761.  761.  775. <cpl>    <cpl>
8 ARIMA(value ~ pdq(1, 1, 2))     1.06   -361.  730.  730.  744. <cpl>    <cpl>
9 ARIMA(value ~ pdq(2, 1, 2))     1.06   -361.  731.  732.  749. <cpl>    <cpl>
```

`ts1` - final model

Truth:

```
ts1 = arima.sim(n=250, model=list(order=c(0,1,2), ma=c(0.4,0.5)))
```

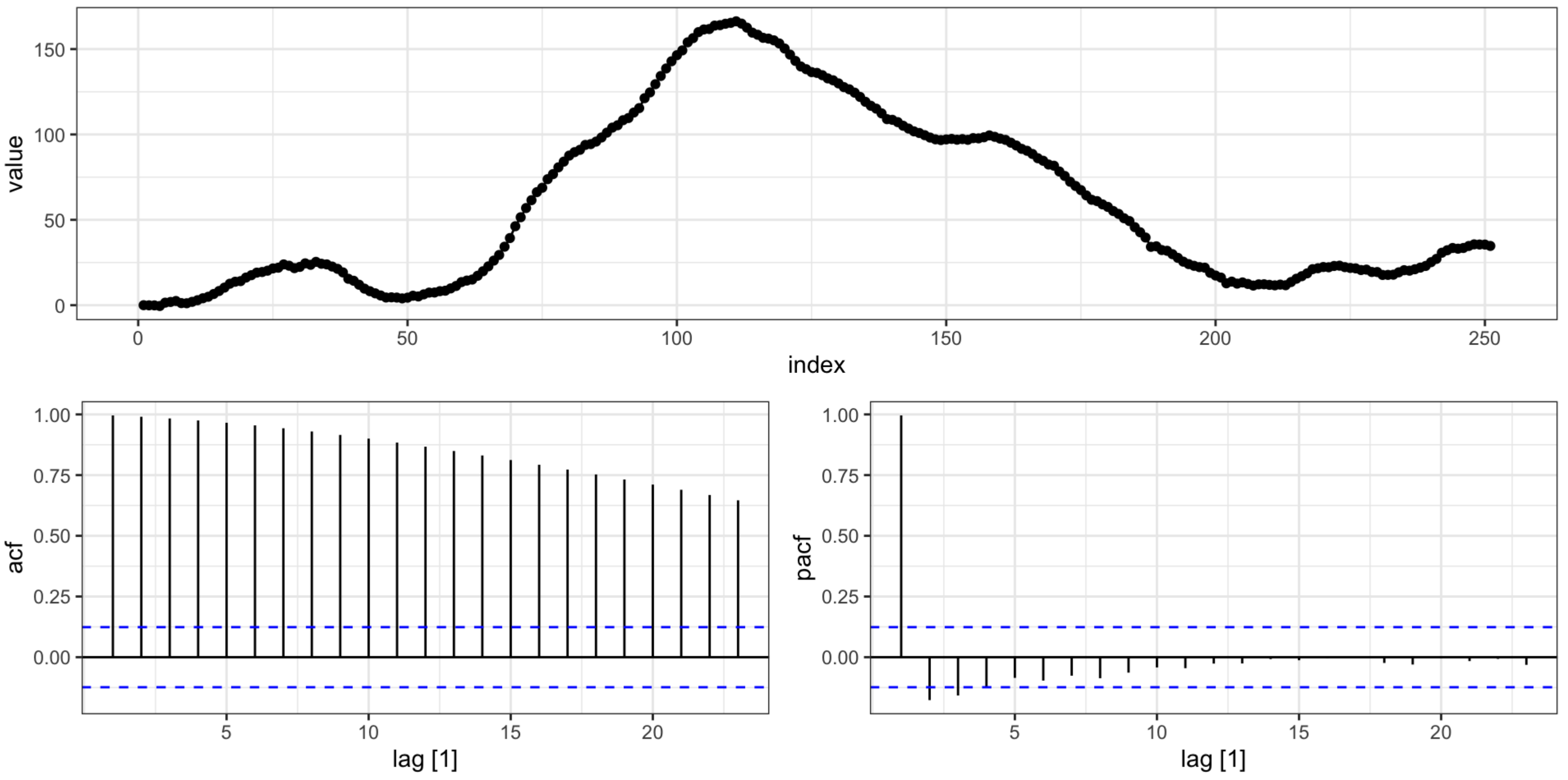
Fitted:

```
model(ts1, final = ARIMA(value ~ pdq(0,1,2))) |>
  report()
```

```
Series: value
Model: ARIMA(0,1,2)

Coefficients:
         ma1     ma2
      0.4328  0.6085
s.e.  0.0526  0.0463

sigma^2 estimated as 1.057:  log likelihood=-361.12
AIC=728.23   AICc=728.33   BIC=738.79
```

ts2

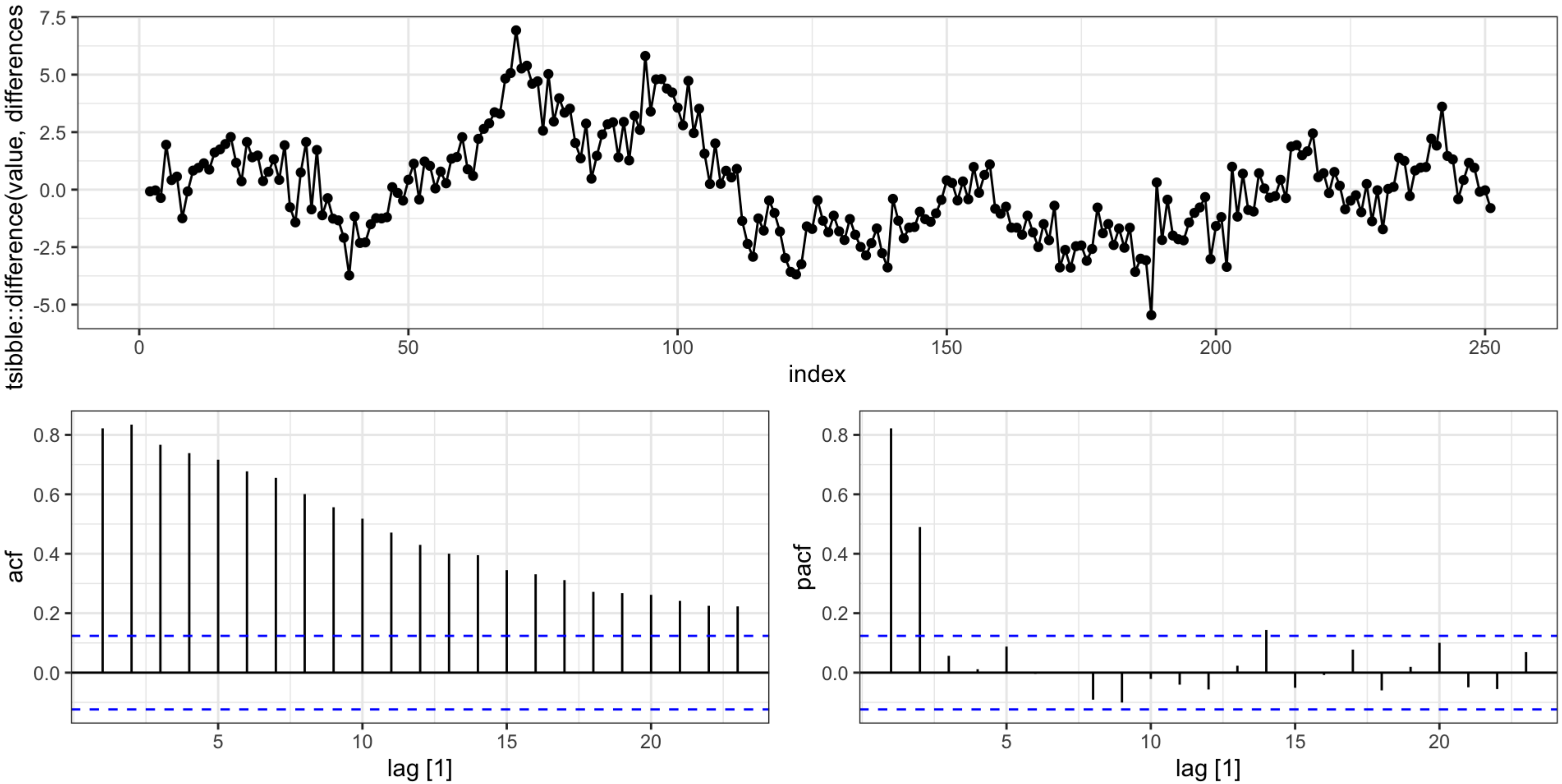
```
feasts::gg_tsdisplay(ts2, y=value, plot_type = "partial")
```



`ts2` - Finding d

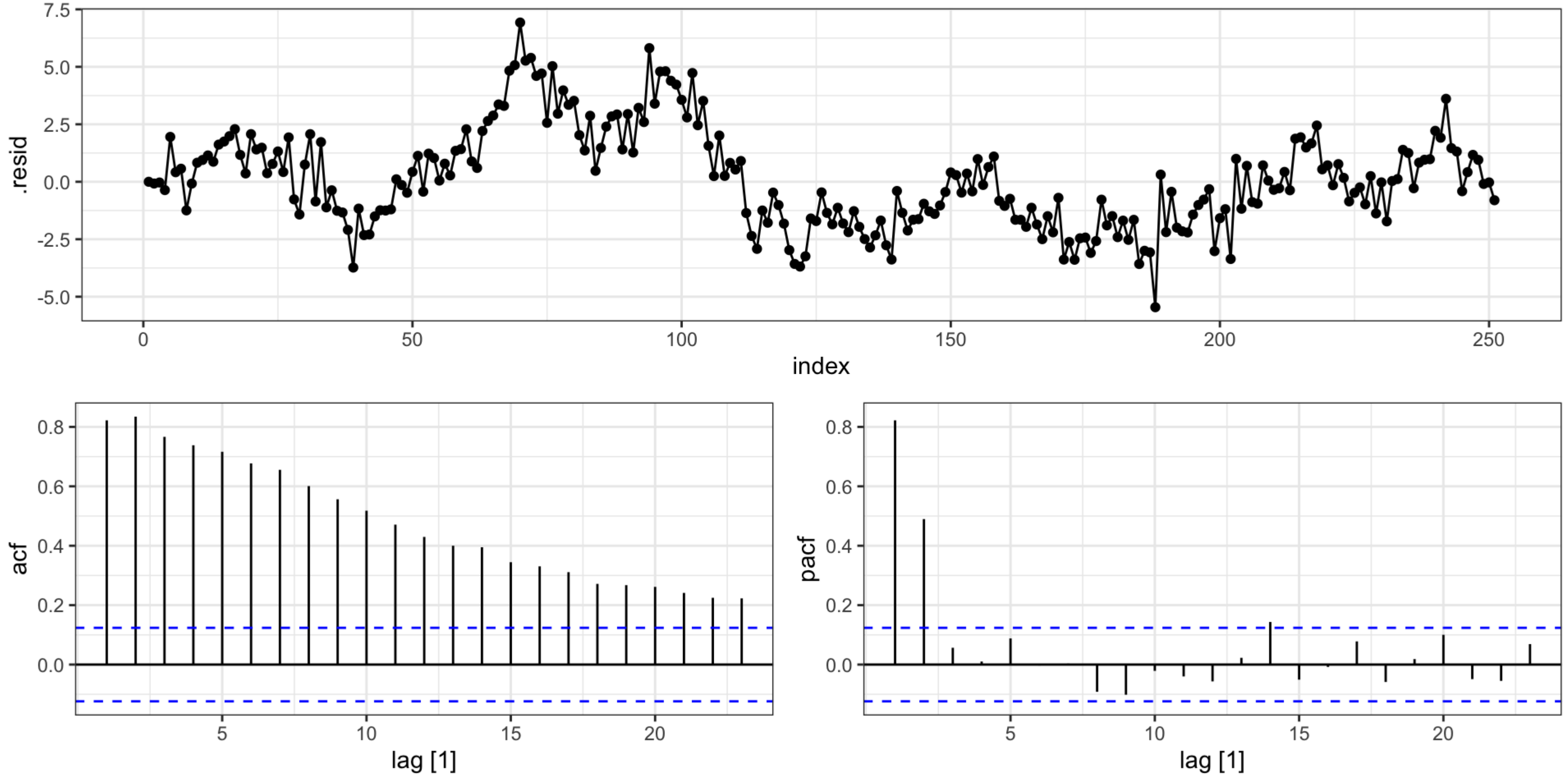
differences = 1 | differences = 2 | differences = 3

```
feasts::gg_tsdisplay(ts2, y=tsibble::difference(value, differences = 1), plot_type = "partial")
```



Residuals - ts2 - ARIMA(0,1,0)

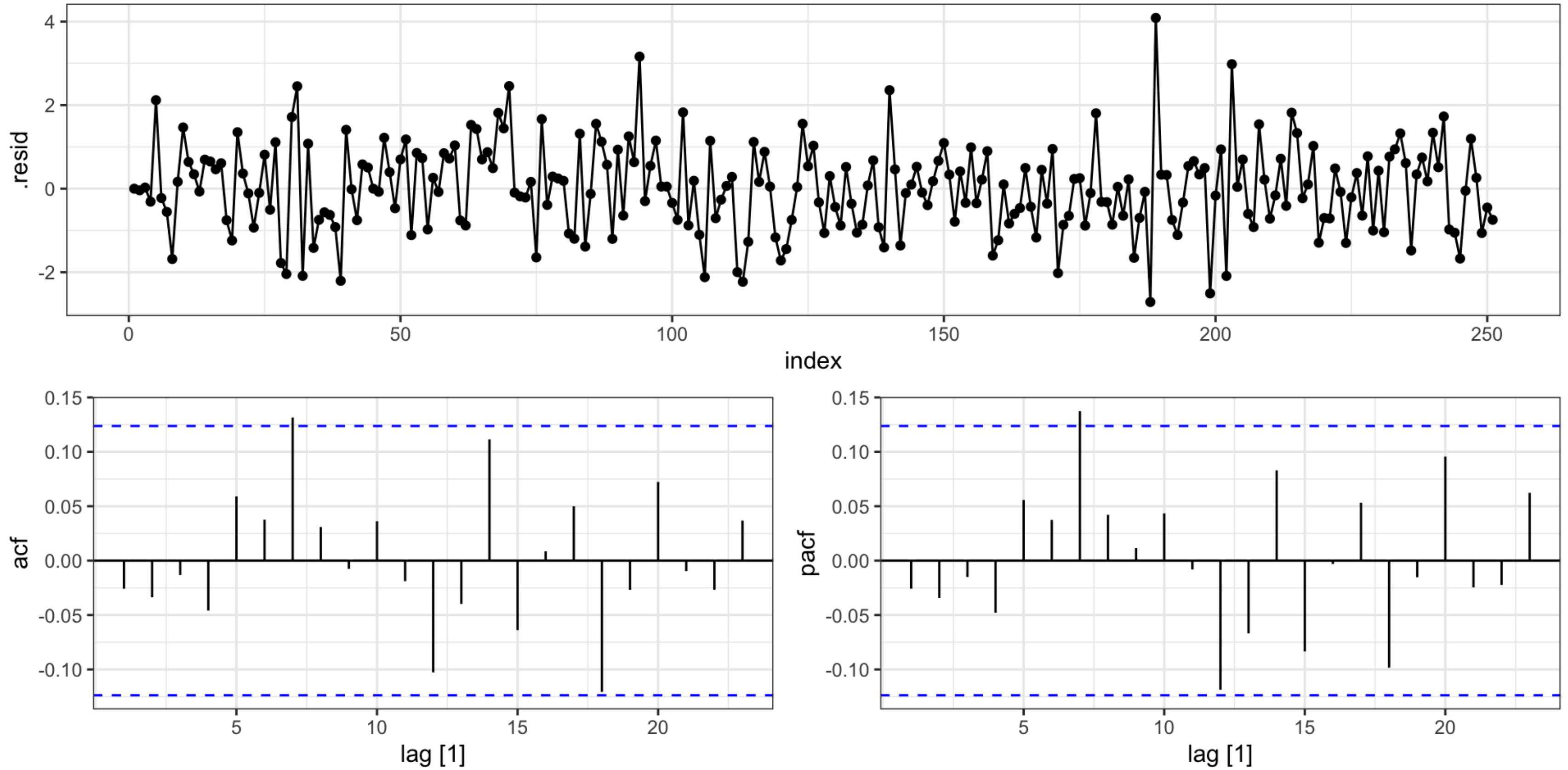
```
model(ts2, ARIMA(value ~ pdq(0,1,0))) |>
  residuals() |>
  feasts::gg_tsdisplay(y = .resid, plot_type = "partial")
```



Residuals - `ts2` - `ARIMA(2,1,0)`

::: {.small}

```
model(ts2, ARIMA(value ~ pdq(2,1,0))) |>
  residuals() |>
  feasts::gg_tsdisplay(y = .resid, plot_type = "partial")
```



`ts2` - Model comparison

```r
model(
  ts2,
  ARIMA(value ~ pdq(0,1,0)), ARIMA(value ~ pdq(1,1,0)), ARIMA(value ~ pdq(0,1,1)),
  ARIMA(value ~ pdq(1,1,1)), ARIMA(value ~ pdq(2,1,0)), ARIMA(value ~ pdq(0,1,2)),
  ARIMA(value ~ pdq(2,1,1)), ARIMA(value ~ pdq(1,1,2)), ARIMA(value ~ pdq(2,1,2))
) |>
  glance()
```

```
# A tibble: 9 × 8
  .model                     sigma2 log_lik   AIC  AICc   BIC ar_roots ma_roots
  <chr>                       <dbl>   <dbl> <dbl> <dbl> <dbl> <list>   <list>
1 ARIMA(value ~ pdq(0, 1, 0))  4.57   -545. 1091. 1091. 1095. <cpl>    <cpl>
2 ARIMA(value ~ pdq(1, 1, 0))  1.48   -404.  811.  812.  819. <cpl>    <cpl>
3 ARIMA(value ~ pdq(0, 1, 1))  2.84   -485.  973.  973.  980. <cpl>    <cpl>
4 ARIMA(value ~ pdq(1, 1, 1))  1.17   -375.  755.  755.  766. <cpl>    <cpl>
5 ARIMA(value ~ pdq(2, 1, 0))  1.13   -370.  745.  745.  756. <cpl>    <cpl>
6 ARIMA(value ~ pdq(0, 1, 2))  1.94   -437.  880.  880.  890. <cpl>    <cpl>
7 ARIMA(value ~ pdq(2, 1, 1))  1.13   -369.  747.  747.  761. <cpl>    <cpl>
8 ARIMA(value ~ pdq(1, 1, 2))  1.15   -371.  750.  750.  764. <cpl>    <cpl>
9 ARIMA(value ~ pdq(2, 1, 2))  1.13   -369.  748.  749.  766. <cpl>    <cpl>
```

ts2 - final model

Truth:

```
ts2 = arima.sim(n=250, model=list(order=c(2,1,0), ar=c(0.4,0.5)))
```

Fitted:

```
model(ts2, final = ARIMA(value ~ pdq(2,1,0))) |>
  report()
```

```
Series: value
Model: ARIMA(2,1,0)

Coefficients:
         ar1     ar2
      0.4181  0.4866
s.e.  0.0548  0.0548

sigma^2 estimated as 1.129:  log likelihood=-369.69
AIC=745.38   AICc=745.48   BIC=755.95
```

Automatic model selection

`ts1`:

```
model(ts1, final = ARIMA(value)) |>
  report()
```

```
Series: value
Model: ARIMA(0,1,2)

Coefficients:
         ma1     ma2
      0.4328  0.6085
s.e.  0.0526  0.0463

sigma^2 estimated as 1.057:  log
likelihood=-361.12
AIC=728.23   AICc=728.33   BIC=738.79
```

`ts2`:

```
model(ts2, final = ARIMA(value)) |>
  report()
```

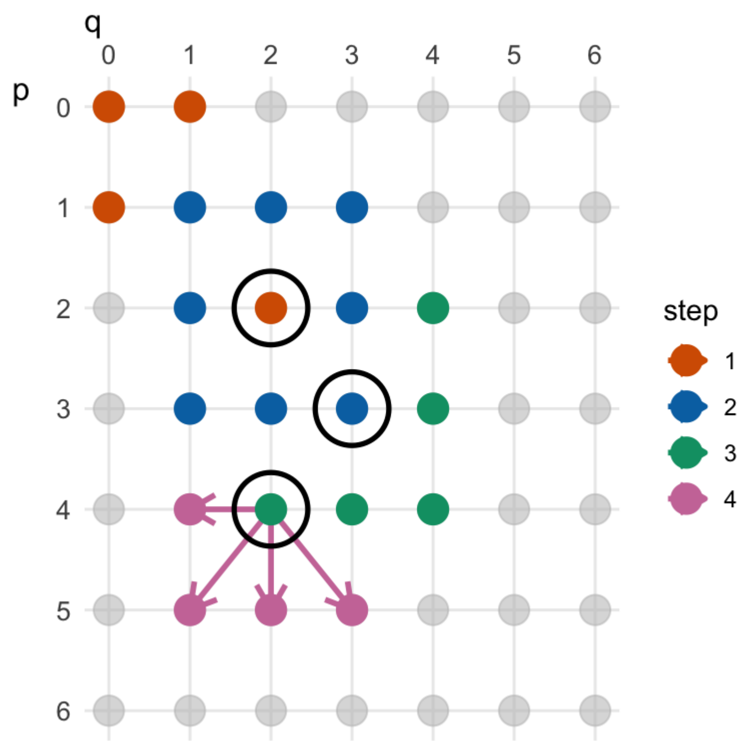
```
Series: value
Model: ARIMA(1,2,1)

Coefficients:
          ar1      ma1
      -0.3977  -0.1896
s.e.   0.1183   0.1320

sigma^2 estimated as 1.158:  log
likelihood=-370.71
AIC=747.42   AICc=747.52   BIC=757.97
```

How does `ARIMA()` work?

- Step 1: Select no. differences `d` via KPSS test
- Step 2: Select current model (with smallest AICc) from:
  - `ARIMA(2, d, 2)`
  - `ARIMA(0, d, 0)`
  - `ARIMA(1, d, 0)`
  - `ARIMA(0, d, 1)`
- Step 3: Consider variations of current model:
  - vary one of `p`, `q`, from current model by ±1;
  - `p`, `q` both vary from current model by ±1;
  - Include/exclude `c` from current model.

  Model with lowest AICc becomes current model.
- Step 4: Repeat Step 3 until no lower AICc can be found.

Search space



From Rob Hyndman's Tidy time series and forecasting in R workshop - Posit::conf 2023

General Guidance

1. Positive autocorrelations out to a large number of lags usually indicates a need for differencing
2. Slightly too much or slightly too little differencing can be corrected by adding AR or MA terms respectively.
3. A model with no differencing usually includes a constant term, a model with two or more orders (rare) differencing usually does not include a constant term.
4. After differencing, if the PACF has a sharp cutoff then consider adding AR terms to the model.
5. After differencing, if the ACF has a sharp cutoff then consider adding an MA term to the model.
6. It is possible for an AR term and an MA term to cancel each other's effects, so try models with fewer AR terms and fewer MA terms.

Based on guidance in Robert Nau's notes on regression and time series analysis - Chapter 5