start(df) *end(df)* *frequency(df)*	when first record was made when last record was made num of records per unit time	*deltat(df)* *time(df)* *cycle(df)*	time increment btw records calculate vector of time indices for observations position in cycle of each observation
diff(..., lag = period)	seasonal diff transformation: remove periodic trends		
ts.plot(cbind(df1, df2))	2 plots on same graph	*diff(log(...))*	generate log returns

Autoregressive Mean-centred version $\phi \in (-1,1)$	$Y_t = c + \phi * Y_{t-1} + \epsilon_t$ Large value of ϕ = greater autocorrelation. Negative values of ϕ = oscillatory time series If $\mu = 0$ and $\phi = 1$, then $Y_t = Y_{t-1} + \epsilon_t$, which is a random walk. Then {Y_t} is not stationary
arima.sim(model = list(ar = phi), n = 50) Persistence = high correlation btw obs and its lag *AR <- arima(df, order = c(1,0,0))*	Simulate AR model Anti-persistence = large variation btw obs and its lag Fit data to AR model (on R, ar1 = $\hat{\phi}$, intercept = $\hat{\mu}$, sigma^2 = $\widehat{\sigma_\epsilon^2}$)
arima.sim(model = list(ma = theta), n = 50) *MA <- arima(df, order = c(0,0,1))* *fitted <- df - residuals(MA)*	Simulate MA model Fit data to MA model (on R, ma1 = $\hat{\theta}$, intercept = $\hat{\mu}$, sigma^2 = $\widehat{\sigma_\epsilon^2}$) $\hat{Y}_t = \hat{\mu} + \hat{\theta}\hat{\epsilon}_{t-1}$. Residuals = $\hat{\epsilon}_t = Y_t - \hat{Y}_t$
MA(1) (e.g. $Y_t = \epsilon_t + \theta\epsilon_{t-1}$) AR(2) (e.g. $Y_t = \phi * Y_{t-2} + \epsilon_t$)	*arima.sim(model = list(order = c(0,0,1), ma = θ), n = 100)* *arima.sim(model = list(order = c(2,0,0), ar = c(0, ϕ)), n = 100)*
acf2(df) *sarima(df, p = .., d = .., q = ..)*	Calculate ACF and PACF pairs Fit data to model
sarima.for(df, n.ahead = .., p, d, q)	Forecasting
acf2(df, max.lag = 60) *sarima(df, 0, 0, 0, P = .., D = .., Q = .., S = ..)*	
Mixed seasonal model *lx <- log(x). dlx <- diff(lx). ddlx <- diff(dlx, 12)* *sarima.for(df, n.ahead = .., p, d, q, P, D, Q, S)*	SARIMA(p, d, q) × (P, D, Q)$_S$ model Log to standardise var. diff to remove trend (but still have seasonal behavior). diff again to get stationary. All 3 = {d = 1, D = 1} Forecasting
Simple exponential smoothing $\hat{y}_{t+h\mid t} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \cdots$	Use all obs for forecast with more recent obs having higher weights *fc <- ses(df, h = 5)* *summary(fc)*Choose α and l_0 by minimizing SSE = $\sum_{t=1}^{T}\left(y_t - \hat{y}_{t\mid t-1}\right)^2$
Holt's linear trend *df %>% holt(h = 5) %>% autoplot*	Small β^* = slope hardly change, so linear trend. High β^* = slope change rapidly ⇒ nonlinear trend Choose α, β^*, l_0, b_0 by minimizing SSE
Damped trend method - allows trend to dampen over time, s.t. it levels off to a constant value *holt(df, damped = TRUE, h = 5)*	$\hat{y}_{t+h\mid t} = l_t + (\phi + \phi^2 + \cdots + \phi^h)b_t$. $l_t = \alpha y_t + (1-\alpha)(l_{t-1} + \phi b_{t-1})$. $b_t = \beta^*(l_t = l_t - 1) + (1-\beta^*)\phi b_{t-1}$. Damping param: 0 < ϕ < 1 Larger ϕ = less damping ⇒ short run forecasts are trended, long run forecasts are constant
BoxCox.lambda(df). ets(df, lambda)	Find estimate of lambda to stabalise var using BoxCox transformation
ARIMA(p, d, q, include.constant = TRUE) *auto.arima(df)* - auto.arima based on Hyndman-Khandakar algo ARIMA(p, d, q, P, D, Q, M) *auto.arima(df, lambda, stepwise, stationary)*	I: Integrated (opp of differencing). d = num of times ts needs to be differenced to make it stationary Selects p and q by minimizing AIC$_c$ value. Select d via unit root tests. Estimate params using MLE - AIC$_c$ can only be compared btw model of same class (ARIMA/ETS only), & same amt of differencing p/q = num of ordinary AR/MA lags. d = num of lag-1 diff. P/Q = num of seasonal AR/MA lags D = num of seasonal diff. m = num of obs per year. lambda for Box-Cox transformation. stepwise = FALSE (to search for more models). stationary = TRUE

ls1 <- list(A = seq(1, 5, by = 2), B = seq(1, 5, length = 4)) ... ls1$A[2] *x <- c(1,2,3), y <- c("1", "2", "3"). df <- data.frame(x, y) ... df$x. df[, "x"]. df[c(3,2),]* *read.csv(). head(). tail(). summary(). ls(). length(). seq(). mean(). median(). sd(). var().* *library(tidyverse); tbl <- as_tibble(df)* *filter(tbl, condition1, condition2, ...). rename(new = old, new2 = old2...)* *select(tbl, col2: col4). select(tbl, !c(col2: col4)). select(tbl, last_col(offset = 1):last_col())* *?function. or help(function)* *tbl %>% filter(cond1) %>% select(col2:col4)* *ggplot(tbl) + geom_point(mapping=aes(x = col1, y = col2)*	creates list creates dataframe common functions tibble filter, rename select, mutate, arrange Get R documentation piping	```{r} # Write R code ```	R markdown Code chunks

X, Y indep: E(XY) = E(X)E(Y)	Var(X) = E(X - μ)^2. E(X) = μ	If a is a constant, Cov(a, X) = 0	Cov(X, X) = Var(X)	Cov(aX, bY) = abCov(X, Y)
Corr(X, Y) = $\frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}}$	If X, Y indep, then Cov(X, Y) = 0	Let U = $\sum_{i=1}^{m} a_i X_i$, V = $\sum_{j=1}^{r} b_j Y_j$, then Cov(U, V) = $\sum_{i=1}^{m}\sum_{j=1}^{r} a_i b_j Cov\left(X_i Y_j\right)$		
Cov(X, Y) = Cov(Y, X) = E[(X - μ_X)(Y - μ_Y)] = E(XY) - $\mu_X\mu_Y$ = $\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{n-1}$	Var($\sum_{i=1}^{n} X_i$) = $\sum_{i=1}^{n} Var(X_i) + 2\sum_{i<j}\sum Cov(X_i, X_j)$			

f(x) = $\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$ Taylor series expansion e^x	Gaussian dist pdf. X ~ N(μ, σ^2). If both X, Y are jointly Gaussian, then they are indep iff Cov(X, Y) = 0 f(x) = $f(a) + \frac{f'(a)}{1!}(x-a)) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots$ 1 + x + $\frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots$. ln(1 + x) = x - $\frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots$. $\frac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots$

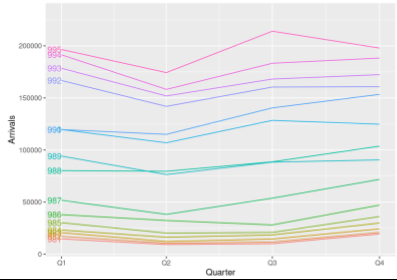
ts <- tsibble(*year = 2015:2019, y = c(123,39,23,32,11)* *index = year)* *q <- seq(as.Date("2016-01-01"), as.Date(2016-12-31"), by = "1 day")* By Year*: Use integers in R* Quarter*: yearquarter(q)* *filter(ts, col1 %in% c("123", "456")) %>%* *autoplot(.vars=y) + geom_point() +* *scale_color_discrete(labels=c("key1", "key2 ")) +* *labs(title = "", y = "", x = "")*	Create tibble w time component Types of col in Tsibble: 1. measurement/obs/record, 2. Index, 3. Key Creates datetime seq. by = day/week/month/quarter/year For Index column. Also have Monthly: *yearmonth()*, Weekly: *yearweek()*. Daily: *as_date()*, *ymd()*. Sub-daily *as_datetime()*, *ymd_hms()* Must only have 1 index col in tsibble. Can have multiple measurement/key cols
Time Series Patterns	Trend: trend exists when there is a long-term incr/decr in data. Does not have to be linear

When looking at graph, consider:
1) Is there a trend? Linear? Change over time?
2) Seasonal effect? Period?
3) Sudden dips/spikes? When?
4) Non-constant variance?

Level: height of series on the ordinate axis
Seasonal: seasonal pattern exists when a series is influenced by factors like quarters/month/day/time. (For ST4253, seasonality always fixed and known period: monthly have period of 12, quarterly 4)
Cycle: rise and fall not of fixed period

Seasonal plot
filter(ts, col1=="xxx",
Quarter <= yearquarter("1995 Q4")) %>%
gg_season(y=yyy, labels="left")

Similar to time plot, except time plot is chopped up into individuals periods, aligned and plotted on an axes for a single period
Compare same period across years
See if similar pattern occur across years
E.g. Arrivals in Q2 generally lower than other quarters
Lines are close to horizontal = no seasonal effect

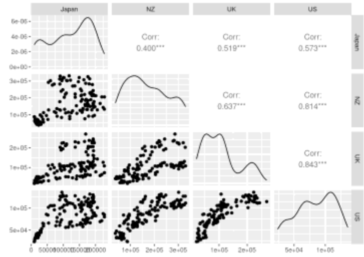


Subseries plots (study seasonal effects)
gg_subseries(ts, y = yyy)

Find patterns within each season
Q1 plot | Q2 plot | Q3 plot | Q4 plot

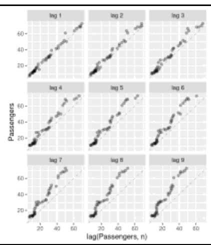
Scatterplots
library(GGally)
pivot_wider(ts, names_from = "Origin", values_from="Arrivals") %>%
ggpairs(columns=2:5)

Diagonal = density plots
Scatterplot w diagonal pattern suggests 2 time series are similar (e.g. UK-US)
Can also see strongest correlation is for US-UK pair
Can use UK arrivals to predict US arrivals



Lag plots
Scatter plot by plotting y_{t-h} on the abscissa (horizontal axis) and y_t on the ordinate(vertical axis)
gg_lag(ts, geom = "point", alpha = 0.3, y = yyy)

For lag 1,2,3,4 there is a strong linear r/s, i.e. to predict y_{t+1}, can use y_t, y_{t-1}, y_{t-2}, y_{t-3}



White Noise (corr = cov = 0)
On graph, adjacent points don't tell us anything about neighbouring points
Sample ACF
ACF(ts) %>% autoplot()
If most vertical lines are close to 0 and within dashed lines (95% CI) = true autocorrelations are all 0, i.e. series is WN

Consider a ts that consists of *uncorrelated* r.v. $\{e_t\}$, s.t. for all t ≥ 1, $E(e_t) = 0$, $Var(e_t) = \sigma_e^2$
This is known as White Noise(WN), $e_t \sim WN(0, \sigma_e^2)$
Special WN: $e_t \sim N(0, \sigma_e^2)$ for all t. This is iid Gaussian, so it is aka Gaussian WN.
r_h = estimate of correlation btw y_t and y_{t-h} = $\frac{\sum_{t=h+1}^{T}(y_t-\bar{y})(y_{t-k}-\bar{y})}{\sum_{t=h+1}^{T}(y_t-\bar{y})^2}$, h = 0, 1, 2, ...
where T is length of series, $\bar{y}$ is sample mean of T observations
r_h is to estimate correlation btw y_t and y_{t-h}, ρ_h, the correlation at lag h

Theoretical ACF for GWN
Let $\gamma_{t,t-h}$ be autocovariance fn for GWN, i.e. $\gamma_{t,t-h} = Cov(e_t, e_{t-h})$

When h = 0, $\gamma_{t,t} = Cov(e_t, e_t) = Var(e_t) = \sigma_e^2$. When h ≥ 1, $\gamma_{t,t-h} = Cov(e_t, e_{t-h}) = 0$
So if we let $\rho_h = Corr(e_t, e_{t-h})$, then $\rho_h = \begin{cases} 1, & h = 0 \\ 0, & h \geq 1 \end{cases}$

Random walk: $y_t = y_{t-1} + e_t$, where $e_t \sim WN(0, \sigma_e^2)$, and $y_0 = 0$
Var increase over time
Covariance also not constant

r_{10} is not a good estimate of $Cov(y_{10}, y_{20})$

$y_1 = y_0 + e_1 = e_1$. ... $y_t = \sum_{i=1}^{t} e_i$. $\rho_{s,t} = Corr(y_s, y_t) = Cov(y_s, y_t) / \sqrt{var(y_s)var(y_t)}$
$E(y_t) = E(\sum_{i=1}^{t} e_i) = \sum_{i=1}^{t} E(e_i) = 0$. $Var(y_t) = var(\sum_{i=1}^{t} e_i) = \sum_{i=1}^{t} var(e_i) = t\sigma_e^2$ (since e_t uncorrelated)
For s ≥ t ≥ 1, $cov(y_s, y_t) = Cov(\sum_{i=1}^{s} e_i, \sum_{i=1}^{t} e_i) = t\sigma_e^2 = \min\{s,t\}\sigma_e^2$
$Corr(y_s, y_t) = t\sigma_e^2 / \sqrt{t\sigma_e^2 \times s\sigma_e^2} = \sqrt{t/s}$
Let t = s-1, then as s → ∞, $\lim_{s\to\infty} \sqrt{t/s} = 1$
1) r = estimate of correlation assume cov btw y_{10} and y_{20} is same as y_{20} and y_{30} and ...
2) $var(y_{20}) \neq var(y_{10})$

Sample ACF is used for:
ACF is +ve when obs are both above/below mean. ACF is -ve when obs are on opp side of mean

1) Check if residuals are WN (close to 0 and btw blue lines)
2) Indicate if there is a strong trend remaining in the data
3) Indicate seasonality, if any
4) If dies down relatively quickly, suggest ARIMA models are appropriate

Tut

Let $e_t \sim GWN(0, \sigma^2)$ and $y_t = e_t e_{t-1}$
So $\gamma_{s,t} = \begin{cases} \sigma^4, & s = t \\ 0, & otherwise \end{cases}$

Mean function $\mu_t = E(y_t) = E(e_t)E(e_{t-1}) = 0$. ACVF = $\gamma_{s,t} = Cov(y_s, y_t)$.
When s = t: $\gamma_{t,t} = Cov(y_t, y_t) = Var(y_t) = E[(e_t e_{t-1})^2] - [E(y_t)]^2 = E(e_t)^2 E(e_{t-1})^2 - 0 = \sigma^4$
When s - t = 1: $\gamma_{t+1,t} = Cov(y_{t+1}, y_t) = Cov(e_{t+1}e_t, e_t e_{t-1}) = E(e_{t+1}e_t e_t e_{t-1}) - E(e_{t+1}e_t)E(e_t e_{t-1}) = 0$

Random walk w drift: $y_t = \delta + y_{t-1} + e_t$, $y_0 = 0$. $E(y_t) = \delta t$

Population adjustments
covid_data <- readRDS(".....rds") %>%
filter(between(date, ymd("2021-01-01"), ymd("2021-12-31")))

Track data per person or per 1000 people...
autoplot(covid_data, .vars = patients_per_million) + labs(x = , y =) +
scale_color_discrete(labels = c("key1", "key2"))

Calendar adjustments
gg_season(ts, y=obs, labels = "left")
mutate(ts, adj_y = obs/days_in_month(index))
%>% gg_season(y = adj_obs, labels = "left")
Trading day variation
Holiday variation

Remove effect of num of days in month, weekend effects, ... to make data simpler
See that obs are lower in months w < 31 days. This causes jagged pattern at troughs on graph
To rectify diff in obs due to num of days in month

Due to changing num of times each day of the week occurs in a mth
Due to presence/absence of a holiday in a mth

To adjust for trading day variation:
Let $\{y_t\}$ be a monthly TS that is a total of some variable for each mth.

We assume $y'_t = T_t + S_t + C_t + e_t$ (trend effect + seasonal effect + trading day effect + noise)
where $C_t = \sum_{i=1}^{7} \alpha_i d_{it}$, where $\sum_{i=1}^{7} \alpha_i = 0$ and d_{it} = fraction of mth t that is day i * 30.4375
To estimate α_i: 1) Decompose y'_t into $y'_t = T_t + S_t + e'_t$. Obtain $w_t = y'_t - \hat{T}_t - \hat{S}_t$

Let $y'_t = (y_t$ / num of days in mth t) * 30.4375 (from 365.25/12)	2) Regress w_t on $d_{1t}, d_{2t}, \dots, d_{7t}$. Obtain estimates $\hat{\alpha}_1, \dots, \hat{\alpha}_7$ 3) Calendar adjusted time series is $y''_t = y'_t - \sum_{i=1}^{7} \hat{\alpha}_i d_{it}$
Inflation adjustments In SG, we have a monthly price index, Consumer Price Index (CPI)	Price index for year t_1 relative to year $t_2 = \frac{price\ of\ set\ of\ items\ in\ year\ t_1}{price\ of\ same\ set\ of\ items\ in\ year\ t_2} * 100$ Adj price at time t_1 relative to time $t_2 = \frac{unadjusted\ price\ in\ year\ t_1}{CPI\ for\ year\ t_1\ relative\ to\ year\ t_2} * 100$
Transformations $w_t = \sqrt{y_t}$, $w_t = \sqrt[3]{y_t}$, $w_t = \log(y_t)$ Box-Cox Transformations Suppose $y_t > 0\ \forall\ t$ Using L'Hopital's rule Let $g(\lambda) = (y_t^{\lambda} - 1)/\lambda$ *l1 <- guerrero(ts$obs)* *mutate(ts, transformed = box_cox(y, l1)) %>% autoplot(.vars = transformed)*	Use if TS has diff variation/variance at diff levels of the TS Log transformation are more interpretable: if $\log_{10}$ is used, an increase of 1 on log scale = multiplication of 10 on original scale $w_t = \begin{cases} log(y_t) & \lambda = 0 \\ (sign(y_t)\lvert y_t \rvert^{\lambda} - 1)/\lambda & \lambda \neq 0 \end{cases}$, λ = 1, 1/2, 1/3, 0, -1 (linear transformation, square root + linear, cube root + linear, natural log, inverse transformation) $\lim_{\lambda \to 0} g(\lambda) = \lim_{\lambda \to 0} \frac{\frac{d}{d\lambda}(y_t^{\lambda} - 1)}{\frac{d}{d\lambda}\lambda} = \lim_{\lambda \to 0} \frac{\frac{d}{d\lambda}\left(e^{\log y_t^{\lambda}} - 1\right)}{1} = \lim_{\lambda \to 0} \frac{d}{d\lambda}\left(e^{\lambda \log y_t}\right) = \lim_{\lambda \to 0} \log y_t\ *\ \left(e^{\lambda \log y_t}\right) = \log y_t$ Compute optimal value of λ to use Intuitive understanding: w = h(y), Var(w) = Var(h(y)) ≈ [h'(E(y))]² * Var(Y) (which we want to find a λ for Var(w) to be constant)

Back-transforming Forecasts $y_t = \begin{cases} exp(w_t) & \lambda = 0 \\ (\lambda w_t + 1)^{1/\lambda} & \lambda \neq 0 \end{cases}$ To reverse transformations to obtain forecasts on original scale f(w) is pdf of w μ_w = E(w) ∫ f(w) dw = 1 ∫ w * f(w) dw = E(w) = μ_w ∫ (w - μ_w)² f(w) dw = E[(w - μ_w)²] = Var(w)	Want to estimate E(y_t) and no transformation used. We use $\hat{y}_t$ to estimate E(y_t) and estimator is unbiased, i.e. E($\hat{y}_t$) = E(y_t) Problem w using transformations is that the back-transformed forecast ≠ the mean, but the median of the forecast dist.
	Suppose we use transformation $w_t = \log(y_t)$. So we expect $y_t = \exp w_t$. Then using estimate, $\widehat{w}_t$, and get $\hat{y}_t = \exp \widehat{w}_t$. Using Taylor's expansion, h(w) = h(a) + h'(a)(w-a) + h''(a) $\frac{(w-a)^2}{2!}$ + ..., $e^w = e^{\mu_w} + e^{\mu_w}(w - \mu_w) + e^{\mu_w}\frac{(w-\mu_w)^2}{2!} + \cdots$ However, E(y_t) = E(e^w) = ∫e^wf(w) dw ≈ $\int \left[e^{\mu_w} + e^{\mu_w}(w - \mu_w) + e^{\mu_w}\frac{(w-\mu_w)^2}{2!}\right] f(w)\, dw = \int e^{\mu_w} f(w)\, dw + \int e^{\mu_w}(w - \mu_w) f(w)\, dw + \int e^{\mu_w}\frac{(w-\mu_w)^2}{2!} f(w)\, dw = e^{\mu_w} + 0 + \frac{1}{2}e^{\mu_w}\int (w - \mu_w)^2 f(w) = e^{\mu_w} + \frac{1}{2}e^{\mu_w}\sigma_w^2 = e^{\mu_w}\left[1 + \frac{1}{2}\sigma_w^2\right]$ So E(y_t) ≠ E(exp $\widehat{w}_t$), i.e the back-transformed estimate is biased.
	So if $\widehat{w}_t$ estimates μ_w, then formula below estimates E(y_t) for $\lambda = 0$. (proof above only shows for $\lambda = 0$) To adjust for this bias, use back transformation: $y_t = \begin{cases} e^{w_t}\left[1 + \frac{\sigma^2}{2}\right] & \lambda = 0 \\ (\lambda w_t + 1)^{1/\lambda}\left[1 + \frac{\sigma^2(1-\lambda)}{2(\lambda w_t + 1)^2}\right] & \lambda \neq 0 \end{cases}$ where σ^2 is the var of w_t.
	For $\lambda \neq 0$: let $w = \frac{1}{\lambda}(y^{\lambda} - 1)$. Then (naive) back-transform = $y = h(w) = (\lambda w + 1)^{1/\lambda}$ $h'(w) = (1/\lambda)(\lambda)(\lambda w + 1)^{1/\lambda - 1} = (\lambda w + 1)^{1/\lambda - 1}$. $h'(w) = (1/\lambda - 1)(\lambda)(\lambda w + 1)^{1/\lambda - 2} = (1 - \lambda)(\lambda w + 1)^{1/\lambda - 2}$ Taylor expansion: $(\lambda w + 1)^{1/\lambda} \approx (\lambda \mu_w + 1)^{1/\lambda} + (\lambda \mu_w + 1)^{1/\lambda - 1}(w - \mu_w) + (1 - \lambda)(\lambda \mu_w + 1)^{1/\lambda - 2}\frac{(w-\mu_w)^2}{2!} + \cdots$ Then E(y) = $\int (\lambda w + 1)^{1/\lambda} f(w)\, dw \approx \int [(\lambda \mu_w + 1)^{1/\lambda} + (\lambda \mu_w + 1)^{1/\lambda - 1}(w - \mu_w) + (1 - \lambda)(\lambda \mu_w + 1)^{1/\lambda - 2}\frac{(w-\mu_w)^2}{2}] f(w)\, dw = (\lambda \mu_w + 1)^{1/\lambda} + 0 + \frac{1}{2}(1 - \lambda)(\lambda \mu_w + 1)^{1/\lambda - 2}\sigma_w^2 = (\lambda \mu_w + 1)^{1/\lambda}\left[1 + \frac{(1-\lambda)\sigma_w^2}{2(\lambda \mu_w + 1)^2}\right]$

Features Summary statistics: *features(ts, obs, quantile)* Tiled statistics: *features(ts, obs, features = feature_set(tags="tile"))* Roll statistics: *features(ts, obs, features = feature_set(tags="roll"))*	Quick comparison btw multiple TS Quantiles for indication of spread and symmetry of nums TS is divided into non-overlapping windows. Mean and var are computed for each window. Var of window means = how "stable" series is. Var of window var = how "lumpy" series is Roll statistics computed using overlapping windows. Used to identify where TS had sharp changes in level, variability and distribution. Shift_level_index = index in TS w largest shift in level. Shift_var_index = index in TS w largest shift in var

Benchmark Forecasting Mtds h = forecasting horizon *library(fpp3)* *report(mtd)* *glance(mtd)* *tidy(mtd)*	1. Average Mtd: Forecast of all future values = mean of historical data, i.e. $\hat{y}_{T+h\lvert T} = \frac{y_1 + y_2 + \cdots + y_T}{T}$ *mean_mtd <- model(ts, avg = MEAN(y)); report(mtd)* # look at model summaries (also have others on LHS) *forecast(mean_mtd, h = 10) %>% autoplot(data = ts, level=NULL)* # level for prediction intervals
	2. Naive Mtd: Forecast = most recent observation, i.e. $\hat{y}_{T+h\lvert T} = y_T$, where e_t ~ WN(0, σ_e^2) and y_0 = 0 *mean_and_naive <- model(ts, avg = MEAN(y), naive = NAIVE(y))* # This creates a tibble w 2 models, MEAN & NAIVE # NAIVE can be replace by *RW()*: random walk w/o drift, since $y_t = y_{t-1} + e_t$, or $y_{T+h} = y_T + \sum_{k=1}^{h} e_{T+k}$. So E($y_{T+h}$| y_1,…,y_T) = y_T *forecast(mean_and_naive, h=10) %>% autoplot(data = ts, level=NULL)* # will product 2 forecasts
Use these mtds as benchmarks to compare w better models	3. Random Walk w Drift: Constant diff btw successive observations apart from the random noise, i.e. $y_t = \delta + y_{t-1} + e_t$ for t ≥ 1, where e_t ~ WN(0, σ_e^2) and y_0 = 0. OR $y_{T+h} = h\delta + y_T + \sum_{k=1}^{h} e_{T+k}$. Since E($y_{T+h}$| y_1, …, y_T) = $h\delta + y_T$ Naive mtd to estimate δ is to use average of diffs btw lag 1 obs, i.e. $\hat{\delta} = \frac{1}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = \frac{y_T - y_1}{T-1}$ Then forecast, $\hat{y}_{T+h\lvert T} = y_T + h\left(\frac{y_T - y_1}{T-1}\right)$ *rwf <- model(ts, rwf = RW(y ~ drift())); forecast(rwf, h=10) %>% autoplot(data = ts, level=NULL)*
# show_gap connects last obs w forecast	4. Seasonal Naive Mtd: Forecast = last observed value from the same season of the previous year Suppose period of time series is m, e.g. m = 12 for monthly data. Then forecast, $\hat{y}_{T+h\lvert T} = y_{T+h-km}$, where k = $\left\lfloor \frac{h-1}{m} \right\rfloor + 1$ *sn <- model(ts, sn = SNAIVE(y)); forecast(sn, h=10) %>% autoplot(data = ts, level=NULL, show_gap=FALSE)*

Residuals: $e_t = y_t - \hat{y}_{t\lvert t-1} = y_t - \hat{y}_t$. Residuals are based on 1-step ahead forecasts.

Suppose we use w_t	Innovation residuals = residuals on the transformed scale	Residuals aka training errors aka Response residuals
$w_t = \log y_t$	$w_t - \widehat{w}_t$	$y_t - \hat{y}_t$, where $\hat{y}_t$ is back-transform of $\widehat{w}_t$ w bias adjustment factor

Extracting Residuals and Fitted Values	*fitted_and_resids <- augment(mean_mtd) # returns a tsibble*
Essential Properties of Residuals	1. Residuals are uncorrelated. If residuals are correlated: ARIMA model might be appropriate. 2. Residuals have mean 0. If mean ≠ 0: mean should be added to forecast to correct for the bias
Desired Properties of Residuals	1. Residuals have constant variance. 2. Residuals are normally distributed These 2 properties make it easier to compute prediction intervals (using Normal Approx). But sometimes, it is impossible to fix them. 1 sol^n is to transform the data
E.g.	*gg_tsresiduals(mean_mtd) # plot TS of residuals + ACF + histogram* If WN: Residuals time plot no autocorrelation. ACF all btw blue line. Histogram looks like Gaussian curve. Ljung Box test have p-value > 0.05.

```
ts <- readRDS("");   l1 <- guererro(ts$y);   snaive <- model(ts, SNAIVE(box_cox(value, l1)))
fcast <- forecast(snaive, h = 12, point_forecast = list(.median=median, .mean=mean)) #OR .median=median(y)
autoplot(fcast, data=filter(ts, index >= yearmonth("1990 Jan")), level=NULL, point_forecast=list(mean=mean), show_gap = FALSE) +
   labs("Bias-adjusted forecast of mean (in red)") +
   autolayer(fcast, level=NULL, point_forecast = list(median=median), color="red")
```

Tests of Autocorrelation

Portmanteau tests: test whether the first h autocorrelations (taken tgt) are significantly diff from what is expected from a WN process.
1) Box-Pierce test.   2) Ljung-Box test

1) Box-Pierce Test: Q = $T\sum_{k=1}^{h} r_k^2$, where r_k = autocorrelation, h is maximum lag being considered, T is num of obs
If each r_k is close to 0, then Q will be small. If some r_k are large, then Q is large, then conclude that residuals are autocorrelated
Rule-of-thumb: Use h = 10 for non-seasonal data. Use h = 2m for seasonal data, where m = period of seasonality.
However, if the h > T/5, then use T/5 instead

2) Ljung-Box test: Q* = $T(T+2)\sum_{k=1}^{h}(T-k)^{-1}r_k^2$. Large value of Q* suggests residuals are not from a WN series.

For both Box-Pierce and Ljung-Box: H_0: $\rho_k = r_k = 0 \;\forall\, k$, i.e. residuals are uncorrelated
Under H_0: both Q and Q* ~ χ^2 dist w (h - K) degrees of freedom, where K = num of parameters in the model
If test is computed based on raw data, then set K = 0

```
augment(mean_and_naive) %>% features(.innov, features=feature_set(tags="portmanteau"), lag=10) #Apply both tests to model
```

Evaluating Forecast Accuracy

Training data = to estimate the parameters of a model. Test data = evaluate its forecast accuracy.
Test set usually 20% of total sample. Test set should ideally be as large as maximum forecast horizon

Forecast error = diff btw observed and its forecast = $e_{T+k} = y_{T+k} - \hat{y}_{T+k|T}$
Training data = {y_1, …, y_T}. Test data = {y_{T+1}, y_{T+2}, …}. Note residuals based on training data, forecast error based on test data

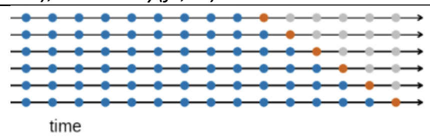
1) Scale-Dependent Errors: computed values, e_{T+k} are on the same scale as the data - Any accuracy measure that is based only on e_{T+k} (instead of a standardisation of it). - Cannot be used to make comparisons btw series of diff scale	MAE (Mean abs error) = $\frac{1}{h}\sum_{k=1}^{h}\lvert e_{T+k}\rvert$. RMSE (root mean squared error) = $\sqrt{\frac{1}{h}\sum_{k=1}^{h} e_{T+k}^2}$

2) Scale-Independent error: To compare forecast errors across diff series w diff scale

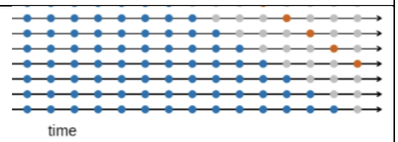
2a) Percentage error = $p_{T+k} = 100 * \frac{e_{T+k}}{y_{T+k}}$ However, if y_{T+k} is close to 0, then p_{T+k} will have extreme values, ∞ p_{T+k} also tend to penalise -ve errors more than positive. If {y_T} is a non-negative series, then for any obs, there is a max positive p_T but negative p_T is unbounded	Mean abs percentage error, MAPE = $\frac{1}{h}\sum_{k=1}^{h}\lvert p_{T+k}\rvert$ symmetric MAPE, sMAPE = $\frac{1}{h}\sum_{k=1}^{h}\frac{200*(y_t-\hat{y}_t)}{y_t+\hat{y}_t}$
2b) Scaled error: Errors are scaled using the training MAE from the naive mtd. (training MAE is acting as baseline) Scaled error < 1 if it is better than naive mtd forecast. Scaled error > 1 if worse than naive mtd forecast - scaled error = $q_j = \frac{e_j}{\frac{1}{T-1}\sum_{t=2}^{T}\lvert y_t - y_{t-1}\rvert}$. For seasonal data, $q_j = \frac{e_j}{\frac{1}{T-m}\sum_{t=m+1}^{T}\lvert y_t - y_{t-m}\rvert}$ (i.e. use seasonal naive for baseline) Both numerator and denominator are on the same scale, hence q_j is scale-indep	Mean abs scaled error, MASE = $\frac{1}{h}\sum_{k=1}^{h}\lvert q_{T+k}\rvert$ i.e. MAE/baseline MAE

```
train <- filter_index(ts, . ~ "Dec 2004");   test <- filter_index(ts, "Jan 2005" ~ .)
models <- model(train, avg=MEAN(y), naive=NAIVE(y))
fc <- forecast(models, h=4);   accuracy(fc, ts)
```

Cross validation



Multi-step cross validation




```
stretched <- stretch_tsibble(ts, .init = 20, .step=1) # generate multiple version of ts, each truncated at the next obs
stretched %>% model(mean = MEAN(Quotes), naive=NAIVE(Quotes)) %>% forecast(h=1) %>%
  filter(Month <= yearmonth("2005 Apr")) %>% accuracy(insurance)
```

Prediction Intervals (PI)

For 95% prediction interval of the next obs = $\hat{y}_t \pm 1.96\hat{\sigma}_h$, where $\hat{\sigma}_h$ is an estimate of the SD of the forecast
When forecasting one step ahead, the SD of the forecast dist ≈ SD of the residuals. , i.e. $\hat{\sigma}_1 \approx \hat{\sigma}$
When there are no parameters estimated, the two SD are identical.
When parameters are estimated, then the SD of the forecast distribution is slightly larger than the residual SD.
As the forecast horizon increases, the prediction intervals generally increase in width. (further ahead, more uncertainty)

Mean	Naive	Seasonal naive	Random Walk w Drift
$\hat{\sigma}_h = \hat{\sigma}\sqrt{1+1/T}$	$\hat{\sigma}_h = \hat{\sigma}\sqrt{h}$	$\hat{\sigma}_h = \hat{\sigma}\sqrt{k+1}$	$\hat{\sigma}_h = \hat{\sigma}\sqrt{h(1+h/(T-1))}$

1) Mean model assumes E(y_t) = μ (which is a parameter), e_t ~ WN(0, σ^2). According to model, actual point in future should be $\mu + e_{T+h}$
$\hat{y}_{T+h|T} = \bar{y}$, where $\bar{y} = \frac{1}{T}\sum_{t=1}^{T} y_t$. $\bar{y}$ is our estimate of μ
But since we estimate μ, our prediction has var = var($\bar{y} + e_{T+h}$) = var($\bar{y}$) + var(e_{T+h}) (indep) = $\frac{T\sigma^2}{T^2} + \sigma^2 = \sigma^2\left(1+\frac{1}{T}\right)$
2) Naive model assumes $y_{T+h} = y_T + \sum_{k=1}^{h} e_{T+k}$. Note y_T is fixed constant here, not a parameter
Variance of prediction = var($\hat{y}_{T+h}$) = var($y_T + \sum_{k=1}^{h} e_{T+k}$) = var($\sum_{k=1}^{h} e_{T+k}$) = h$\sigma^2$

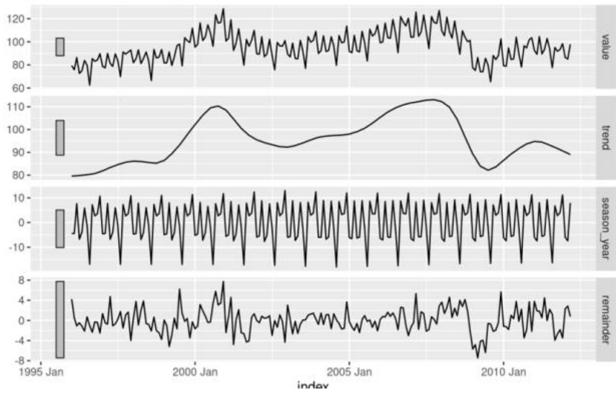
If a transformation has been used, then the prediction interval should be computed on the transformed scale.
End-points of PI should be back-transformed to give a PI on the original scale.
The new intervals will have the same coverage, but they will no longer be symmetric.

```
autoplot(fcast, data=filter(ts, index >= yearmonth("1990 Jan")), level=c(80, 95), #point_forecast=list(mean=mean), show_gap = FALSE)
# level for 80 & 95% PI  # mean=bias adjusted, median = w/o bias adjustment
```

```
## Simulating from the residuals. (if residuals not normal) In contrast to calculating σ̂, the estimate of SD of forecast
fc2 <- eggs_mdl %>%  forecast(h = 50, bootstrap = TRUE) %>%  mutate(.median = median(eggs))
```

Decomposition	Time series can have diff patterns: trends, cycles and seasonal effects Can decompose TS into 3 components: trend-cycle component, seasonal component, remainder component (unable to explain)
Additive Decomposition	$y_t = S_t + T_t + R_t$, where S_t = seasonal component, T_t = trend-cycle component, R_t = remainder component Model is appropriate if seasonal variation does not change w level of TS
Multiplicative Decomposition	$y_t = S_t * T_t * R_t$ Model is appropriate if variation around the trend-cycle appears to be proportional to level of TS

	Can also log $y_t = \log S_t + \log T_t + \log R_t$ to transform multiplicative model to additive model
E.g. To interpret additive decomposition	*dcmp <- TS %>% model(stl = STL(y))* *components(dcmp)* *components(dcmp) %>% as_tsibble() %>% autoplot(y, colour="gray") + geom_line(aes(y=trend), colour="red")* In table, value = y_t, trend = $\hat{T}_t$, season_year = $\hat{S}_t$, remainder = $\hat{R}_t$, season_adjust = $y_t - \hat{S}_t = \hat{T}_t + \hat{R}_t$ See season_adjusted line overlaid on TS
	components(dcmp) %>% autoplot() From top to bottom: y_t, $\hat{T}_t$, $\hat{S}_t$, $\hat{R}_t$ Grey bar on left are of same value across all plots. i.e. Trend and seasonality explains a larger proportion of the series, and remainder explaining less. Note remainder is largest at around spikes/dips in the trend If seasonal variaion is not of primary interest, we should focus on the seasonally adjusted series, i.e. For additive models: $y_t - \hat{S}_t$. For multiplicative model: $y_t / \hat{S}_t$ Since seasonally adjusted data still contain remainder component, it will not be smooth *components(dcmp) %>% as_tsibble() %>% autoplot(y, colour="gray") + geom_line(aes(y=season_adjust), colour="blue")*



Decomposition Features	Strength of trend = $F_t = max\left(0,1 - \frac{Var(R_t)}{Var(T_t+R_t)}\right)$. Small value of F_t (close to 0) indicates $Var(R_t) \approx Var(T_t + R_t)$, i.e. trend is not the "driving force" compared to residual noise. If F_t close to 1, then trend is very impt compared to residual noise Strength of seasonality = $F_S = max\left(0,1 - \frac{Var(R_t)}{Var(S_t+R_t)}\right)$
	features(TS, y, features=feature_set(tags="seasonal") # trend_strength = F_t, *seasonal_strength_year* = F_S. Also have other values below

seasonal_peak_year	timing of peaks within a season
seasonal_trough_year	timing of throughs within a season
spikiness	prevalence of spikes in R_t of the STL decomposition = var of leave-one-out variances of R_t
linearity	linearity of T_t of the STL decomposition. Based on coefficient of a Linear Regression applied to T_t
curvature	curvature of T_t of the STL decomp. Based on coeff from an orthogonal quadratic regression applied to T_t

Moving Average Filters	1st step in TS decomposition is to estimate trend-cycle. Use Moving Average to smooth the input process MA of order m is a special case of a linear filter: $\sum_{j=-k}^{k} \frac{1}{m} y_{t+j}$, where m = 2k+1 is an odd num. This operator aka m-MA m-MA is a simple mtd to estimate trend-cycle component. $\hat{T}_t = \sum_{j=-k}^{k} \frac{1}{m} y_{t+j}$. It averages nearby values to return smoothed version of TS m = known seasonal period, so as to remove seasonal variation. This happens if sum of seasonal components for each period = 0 m-MA aka finite MA filter / finite order linear filter / time invariant linear filter / low-pass filter
E.g. computing MA	*ts2 <- mutate(ts, \`5-MA\` = slide_dbl(y, mean, .before=2, .after=2, .complete=TRUE))* Smoother series capture the main movement of the TS w/o all the minor fluctuations. Larger m = smoother curve
Computations when m is Even	If m is even, convention is to take one obs more from the future than the past. E.g. m = 2k. Then filter = $\sum_{j=-(k-1)}^{k} \frac{1}{m} y_{t+j}$ *ts <- mutate(ts, \`4-MA\` = slide_dbl(y, mean, .before=1, .after=2, .complete=TRUE))* When m = 2k+1 is odd, m-MA operation is symmetric: k earlier & later obs & middle obs. Each obs multiplied by 1/, (symmetric in weights assigned to obs before and after middle one) When m = 2k is even, not symmetric. Repeated application of the MA filter can yield a filter w the symmetry properties we had. Applying a $2 \times m$ - MA means 1) apply m-MA to raw data w extra obs from future. 2) Apply 2-MA to the new col w an extra obs from the past, i.e. 4-MA: $y_t^R = \frac{1}{4}(y_{t-1} + y_t + y_{t+1} + y_{t+2})$. $2 \times$ 4-MA: $y_t' = \frac{1}{2}(y_{t-1}^R + y_t^R) = \frac{1}{8}y_{t-2} + \frac{1}{4}y_{t-1} + \frac{1}{4}y_t + \frac{1}{4}y_{t+1} + \frac{1}{8}y_{t+2}$ Now, num of obs on both sides of t are same. Weights assigned to points same dist from t but on opp sides are the same *ts <- mutate(ts, \`2x4-MA\` = slide_dbl(\`4-MA\`, mean, .before=1, .after=0, .complete=TRUE))* $2 \times m$ - MA aka centred moving average of order m. Note, odd order MA don't have to be centered – already symmetric MA thus help to estimate trend cycle from seasonal data, by removing the seasonal variation
Removing Seasonal Variation using a Filter	Intuitively, seasonal variation is that periodic component which is centred around the trend-cycle component. Hence, reasonable to assume seasonal components sum to 0 E.g. Quarterly data implies $S_t = S_{t+4}$ for all t, and $\sum_{j=0}^{3} S_{t+j} = 0$ for all t. The appropriate filter to use is the $2 \times$ 4-MA $\hat{T}_t = \frac{1}{8}y_{t-2} + \frac{1}{4}y_{t-1} + \frac{1}{4}y_t + \frac{1}{4}y_{t+1} + \frac{1}{8}y_{t+2} = \frac{1}{8}(T_{t-2} + R_{t-2}) + \frac{1}{4}(T_{t-1} + R_{t-1}) + \frac{1}{4}(T_t + R_t) + \frac{1}{4}(T_{t+1} + R_{t+1}) + \frac{1}{8}(T_{t+2} + R_{t+2}) + \frac{1}{8}S_{t-2} + \frac{1}{4}S_{t-1} + \frac{1}{4}S_t + \frac{1}{4}S_{t+1} + \frac{1}{8}S_{t+2} = \frac{1}{8}(T_{t-2} + R_{t-2}) + \frac{1}{4}(T_{t-1} + R_{t-1}) + \frac{1}{4}(T_t + R_t) + \frac{1}{4}(T_{t+1} + R_{t+1}) + \frac{1}{8}(T_{t+2} + R_{t+2}) + \frac{1}{4}\sum_{j=0}^{3} S_{t+j} = \frac{1}{8}(T_{t-2} + R_{t-2}) + \frac{1}{4}(T_{t-1} + R_{t-1}) + \frac{1}{4}(T_t + R_t) + \frac{1}{4}(T_{t+1} + R_{t+1}) + \frac{1}{8}(T_{t+2} + R_{t+2})$
Guidelines	Use $2 \times$ 12-MA for monthly date and 7-MA for daily data
Weighted Moving Averages	Weighted m-MA: $\hat{T}_t = \sum_{j=-k}^{k} a_j y_{t+j}$, where k = (m-1)/2 Simple m-MA is the special case where a_j = 1/m for all j. $2 \times$ 4-MA is another special case, w weights 1/8, 1/4, 1/4, 1/4, 1/8 Weighted MA yield smoother estimates than simple MA since weights are slowly ↑ and ↓, instead of abruptly including/excluding them By choosing appropriate weights, can design filter to remove higher-order terms that are likely noise To design filter: start w window size m, fit a polynomial trend within that window by minimising least squares error. Replace obs in the middle of that window w point on fitted polynomial E.g. want to find weights for a_{-2}, a_{-1}, a_0, a_1, a_2 (i.e. m = 5) by fitting quadratic trend. WLOG, can consider points at t = -2, -1, 0, 1, 2 Fn to minimize = $h(b_0, b_1, b_2) = \sum_{t=-2}^{2}(y_t - b_0 - b_1 t - b_2 t^2)^2$. Only need to estimat b_0 Taking partial derivatives (w.r.t b_0, b_1, b_2) one at a time and setting to 0, we get $\sum y_t = 5b_0 + 10b_2$. $\sum t y_t = 10b_1$. $\sum t^2 y_t = 10b_0 + 34b_2$ Solving for b_0: $b_0 = -\frac{6}{70}y_{-2} + \frac{24}{70}y_{-1} + \frac{17}{35}y_0 + \frac{24}{70}y_1 - \frac{6}{70}y_2$. Weighted 5-MA are $\left[-\frac{6}{70}, \frac{24}{70}, \frac{17}{35}, \frac{24}{70}, -\frac{6}{70}\right]$ Weighted 15-MA: $[a_0, a_1, \ldots, a_7] = \frac{1}{320}[74, 67, 46, 21, 3, -5, -6, -3]$. Allow cubic trend to pass through
E.g. Weighted MA	Creating TS w quadratic trend. *x <- 1:10; y <- ts(x^2) %>% as_tsibble()* To apply weighted MA: *stats::filter(y$value, c(-6/70, 24/70, 17/35, 24/70, -6/70)* Series is unmodified by filter, i.e. filter allows a signal to pass through undistorted
	Classical Additive Decomp. Assume seasonal effect is same for all t

Decomposition algos

1) If m is even, use a $2 \times$ m-MA (if m is odd, use simple m-MA) to compute $\hat{T}_t$
2) Calculate the de-trended series, $y_t - \hat{T}_t$
3) Estimate the seasonal component: a) Average the de-trended values for each month. E.g., average all de-trended March values to obtain the estimate of the effect of the March season.
b) Adjust the seasonal component so that they sum to 0 to get $\hat{S}_t$
c) This last step ensures that there is no confounding of the seasonal effects with the level of the time series, and allows us to view the seasonal effects as deviations from the trend-cycle.
4) Calculate the remainder component using $\hat{R}_t = y_t - \hat{T}_t - \hat{S}_t$
Suppose additive model is appropriate for our series, $y_t = \hat{T}_t + \hat{S}_t + \hat{R}_t$
Whatever we estimate for the trend-cycle and seasonal components, we could always add/subtract an arbitrary value δ to each of them, i.e. $\hat{y}_t = \hat{T}_t + \hat{S}_t = (\hat{T}_t + \delta) + (\hat{S}_t - \delta)$
To avoid this non-identifiability, constraint our seasonal effects to sum to 0. Also allows us to interpret the average seasonal effect as 0.
Suppose we have our initial estimates of the seasonal effects, $\hat{S}_1^0, \hat{S}_2^0, \ldots, \hat{S}_m^0$
Then can adjust these by setting $\hat{S}_t = \hat{S}_t^0 - \frac{1}{m}\sum_{k=1}^{m}\hat{S}_k^0$ for t = 1, …, m

Classical Multiplicative Decomp. Assume seasonal effect is same for all t
1) If m is even, use a $2 \times$ m-MA (if m is odd, use simple m-MA) to compute $\hat{T}_t$
2) Calculate the de-trended series, $y_t / \hat{T}_t$
3) Estimate the seasonal component: a) Average the de-trended values for each month.
b) Adjust the seasonal component so that they sum to m to get $\hat{S}_t$
c) This ensures that the average of the seasonal effects = 1; each is then a multiplicative deviation from the trend-cycle
4) Calculate the remainder component using $\hat{R}_t = \frac{y_t}{\hat{T}_t \hat{S}_t}$
To constraint our seasonal effects to sum to m. Suppose we have our initial estimates of the seasonal effects, $\hat{S}_1^0, \hat{S}_2^0, \ldots, \hat{S}_m^0$
Then can adjust these by setting $\hat{S}_t = \frac{\hat{S}_t^0}{\frac{1}{m}\sum_{k=1}^{m}\hat{S}_k^0}$ for t = 1, …, m

E.g. *ts_dc <- model(ts, class_add = classical_decomposition(y, "additive")) # OR multiplicative*
components(ts_dc) %>% autoplot()

Cons of classical: Since we are using MA filters, we are unable to estimate the trend-cycle for the beginning and end of the series. The classical approach assumes that the seasonal variation is the same over time. It is also not robust to outliers.

X11 Decomp: based on classical decomp, but has some improvements:
- By using one-sided linear filters, it obtains trend-cycle estimates for all time points.
- Allows seasonal effect to vary over time.
- Includes use of a regression model for the remainder component
- Annual holidays are included in the seasonal components.
- The process iterates the algo to achieve smoother estimates.
X11-ARIMA, X12-ARIMA and X13-ARIMA are all improvements on X11
OR y ~ regression(variables='td', aictest=null) #td = trading day variation

```
ts_dc <- model(ts,
    class_add = classical_decomposition(y, "additive"),
    class_mult = classical_decomposition(y, "multiplicative"),
    x11 = X_13ARIMA_SEATS(y ~ x11())  )

select(ts_dc, class_mult) %>%
   components() %>% autoplot()
select(ts_dc, x11) %>%
    components() %>% autoplot()
```

Seasonal and Trend decomposition using LOESS (STL)

"Loess" refers to a locally weighted regression model. This is used in place of a moving average filter for estimating the trend-cycle.
In comparison to the ordinary linear regression model, loess is able to estimate nonlinear relationships.

Advantages of STL	Disadvantages of STL
- Unlike X11, STL can handle any type of seasonality (fixed pattern) - Like X11, it allows seasonal component to change over time (same pattern, but value of pattern changes) - Smoothness of the trend cycle can be controlled by the user. - Can be made robust to outliers, so that occasional unusual observations will not affect estimates of the trend cycle.	- Can only handle additive models. - This shortcoming can be somewhat overcome by transforming model first. - There are several parameters for this approach. - There are defaults for several of them except one.

Loess Fitting. Suppose x_i and y_i are measurements of an indep and dependent variables.
Loess regression curve, $\hat{g}(x)$ is a smoothing of y given x that can be computed for any values of x. To compute $\hat{g}$:
1) Choose a value $q > 0$, that will serve as the span.
2) The q values of x_i that are closest to x will be given a weight, based on how far they are from x, typically through a kernel function.
3) x_i values that are closer to x will receive a larger weight.
4) Perform a weighted least squares regression using the above weights.
geom_smooth(span=%ofpoints, method="loess", method.args=list(degree=0))

STL algo consists of two loops:
- Outer loop for robustness to outliers in the time series. (If sure no outliers, no need outer loop; usually 5-10 iterations)
- Inner loop to estimate trend and seasonal components. Recall that seasonal component can vary over time. (usually 1-2 iterations)
- The loess algo is repeatedly used as the smoother, except in one portion of the procedure.
In the outer loop: {
  1) The remainder component is estimated.
  2) They are assigned a robustness weight (points w larger "residuals" given lower weight), which is used in the inner loop. }
In inner loop, w a given set of robustness weights and an initial trend-cycle estimate: {
  1) Series is detrended using $\hat{T}_t^{(k)}$
  2) Individual subseries are smoothed using loess (*s.window*) to get $\hat{S}_t^{(k)}$. E.g. if data has monthly freq, 12 smoothings are carried out
  3) Combine smoothed subseries into 1, apply MA filter twice. Estimate loess (*l.window*) smoothing for this combined series, $\hat{T}_t'^{(k)}$. (to identify and extracts any residual trend)
  4) $\hat{S}_t^{(k+1)} = \hat{S}_t^{(k)} - \hat{T}_t'^{(k)}$, to yield an estimate of the seasonal component.
  5) New series, $y_t - \hat{S}_t^{(k+1)}$ is smoothed by a loess smoother (*t.window*) to obtain a new trend-cycle estimate, $\hat{T}_t^{(k+1)}$. }

```
model(ts, stl1 = STL(Y ~ trend(window=, degree=1) + season(window=) + lowpass())

dc_stl <- model(ts,
   stl1 = STL(value ~ trend(window=13), robust=TRUE),
```

t.window, s.window, l.window
Usually, just use default for trend and lowpass
t.window: trend cycle window
s.window: seasonal window

<table>
<tr><td></td><td>stl2 = STL(value ~ trend(window=10) + season(window="periodic"), robust=TRUE))
select(dcmp_stl, stl2) %>%
components() %>% autoplot()</td><td>s.window = "periodic" : assume seasonal component don't change.
robust : model not influenced by outliers much</td></tr>
<tr><td></td><td>select(dcmp_stl, stl2) %>% components() %>% as_tsibble() %>% mutate(raw_mth = Month(index, label=TRUE) + ggplot(aes(x=index)) + geom_line(aes(y=season_year)) + geom_point(aes(y=Y-trend)) + facet_wrap(~raw_mth, nrow=4)</td><td>For all subseries, check if s.window needs to be change or not. If line underfit, decr window. Overfit = incr window</td></tr>
<tr><td></td><td colspan="2">season(window = Inf) : seasonality effect same</td></tr>
<tr><td>Forecasting w decomp</td><td>fit_dcmp <- model(ts, dcmp_fc = decomposition_model(
STL(value ~ trend(window=13) + season(window="periodic"), robust=TRUE),
NAIVE(season_adjust), SNAIVE(season_year)))
forecast(fit_dcmp, h=12) %>% autoplot(elecequip)</td><td>In decomposition_model, specify:
- decomposition method,
- forecasting mtd for the seasonally adjusted series,
- forecasting mtd for the season effect.</td></tr>
<tr><td>Tut</td><td colspan="2">Suppose TS is $y_t = f(t) + e_t$, where f is a smooth and cts fn of t, and $e_t \sim$ WN(0, σ^2).
Can use m-MA to estimate f(t), w m = 2k + 1: $\hat{f}(t) = \frac{1}{2k+1}\sum_{j=-k}^{k} y_{t+j}$, t = k+1, …, n-k. i.e. Larger m = smaller var, higher bias
Taylor expansion of f about t: $f(t+j) \approx f(t) + f'(t)(t+j-t) + \frac{1}{2}f''(t)(t+j-t)^2 + \cdots = f(t) + f'(t)\cdot j + \frac{1}{2}f''(t)\cdot j^2 + \cdots$
Then, $\hat{f}(t) = \frac{1}{2k+1}\sum_{j=-k}^{k} f(t+j) + \frac{1}{2k+1}\sum_{j=-k}^{k} e_{t+j}$. Bias = $E[\hat{f}(t)] - f(t) = \frac{k(k+1)}{6}f''(t)$
$E[\hat{f}(t)] \approx f(t) + \frac{1}{2k+1}f'(t)\sum_{j=-k}^{k} j + \frac{1}{2}\frac{1}{2k+1}f''(t)\sum_{j=-k}^{k} j^2 = f(t) + 0 + \frac{1}{2(2k+1)}f''(t)2\frac{k(k+1)(2k+1)}{6} = f(t) + \frac{k(k+1)}{6}f''(t)$
$var[\hat{f}(t)] = var\left[\frac{1}{2k+1}\sum_{j=-k}^{k} e_{t+j}\right] = \frac{1}{2k+1}\sigma^2$</td></tr>
<tr><td></td><td colspan="2">Symmetric MA $\{a_j\}$, j = -q, …, 0, …, q passes an arbitrary polynomial of deg k w/o distortion,
i.e. $m_t = \sum_{j=-q}^{q} a_j m_{t+j}$ ∀ k^{th} deg polynomial $m_t = c_0 + c_1 t + \ldots + c_k t^k$ iff $\sum_j a_j = 1$ and $\sum_j j^r a_j = 0$ for r = 1, …, k
Proof: $m_t = \sum_{i=0}^{k} c_i t^i$, and $m_{t+j} = \sum_{i=0}^{k} c_i (t+j)^i$. Note since MA is symmetric, $a_j = a_{-j}$
RHS = $\sum_{j=-q}^{q} a_j m_{t+j} = \sum_j a_j \sum_{i=0}^{k} c_i (t+j)^i = \sum_{i=0}^{k} c_i \sum_j a_j (t+j)^i$. Now need to show $\sum_{i=0}^{k} c_i \sum_j a_j (t+j)^i = \sum_{i=0}^{k} c_i t^i = m_t$
So just need to show $\sum_j a_j (t+j)^i = t^i$. Using binomial expansion, $(t+j)^i = \sum_{n=0}^{i}\binom{i}{n} t^n j^{i-n}$
$\sum_j a_j (t+j)^i = \sum_j a_j \sum_{n=0}^{i}\binom{i}{n} t^n j^{i-n} = \sum_{n=0}^{i}\binom{i}{n} t^n \sum_j a_j j^{i-n} = \sum_{n=0}^{i-1}\binom{i}{n} t^n \sum_j a_j j^{i-n} + \binom{i}{i} t^i \sum_j a_j j^{i-i}$
$= \sum_{n=0}^{i-1}\binom{i}{n} t^n \sum_j a_j j^{i-n} + t^i \sum_j a_j = \sum_{n=0}^{i-1}\binom{i}{n} t^n * 0 + t^i * 1 = t^i$ (using 2 conditions above)</td></tr>
</table>

<table>
<tr><td>Methods vs Models</td><td>Forecasting mtd = algo that provides a point pred in future. Statistical model = process that generates data - probability dist for future
- Point forecast can then be obtained by taking mean/median of that probability dist
Note $\hat{y}_{t+h} = \hat{y}_{t+h|t}$ = forecast of y_{t+h} given $y_1, \ldots, y_t$ (in-sample). $\hat{y}_{T+h|T}$ = forecast of y_{T+h}, given $y_1, \ldots, y_T$ (out-of-sample)
Mtd e.g. $\hat{y}_{t+h|t} = \bar{y}$ for all h (Mean mtd used to generate forecasts w/o any further assumptions)
Model e.g. $y_t = \mu + e_t$, where $e_t \sim$ GWN(0, σ^2). Model implies $y_{t+h|t} \sim$ N(μ, σ^2) for all h. Once we estimate μ and σ^2, we can use the mean of the estimated dist to forecast y_{t+h}: $\hat{y}_{t+h|t} = \hat{\mu}$. Can also forecast that w Prob 0.95, $y_{t+1} \in (\hat{\mu} - 1.96 * \hat{\sigma}, \hat{\mu} + 1.96 * \hat{\sigma})$
Model allows us to compute prediction intervals (PI). But requires making distributional assumptions
A model fully specifies the data generating process, whereas a forecast mtd does not.</td></tr>
<tr><td>State space models</td><td>Let x_t be a "state vector" containing unmeasured components that describe the level, trend and seasonality of the series
Then a linear innovations state space can be written as $y_t = w'x_{t-1} + e_t$ – (measurement eqn). And $x_t = Fx_{t-1} + ge_t$ – (transition/state eqn), where e_t is a WN process, g and w are vectors and F is a matrix
Coefficient matrices and vectors could contain parameters that need to be estimated, but don't involve state x_{t-1}
This is known as the Innovations formula = assume identical errors in both eqns
Alternative: assume there is 1 error for the measurement eqn, e_t and an independent error, z_t for the transition eqn
Instead of F, g and w being matrices, can also generalised s.t. $y_t = w(x_{t-1}) + r(x_{t-1})e_t$ and $x_t = f(x_{t-1}) + g(x_{t-1})e_t$, where functions w and r take in vector and return scalar, while fns f and g returns a vector</td></tr>
<tr><td>Exponential Smoothing (ETS)</td><td>In exponential smoothing, view trend as a combination of a local level, l and a local growth term b.
Let T_h = forecast of trend h time periods ahead, and ϕ = damping parameter (0 < ϕ < 1)
<table>
<tr><td>Trend type</td><td>None</td><td>Additive</td><td>Additive damped</td><td>Multiplicative</td><td>Multiplicative damped</td></tr>
<tr><td>Formula</td><td>$T_h = l$</td><td>$T_h = l + bh$</td><td>$T_h = l + (\phi + \phi^2 + \cdots + \phi^h)b$</td><td>$T_h = lb^h$</td><td>$T_h = lb^{(\phi+\phi^2+\cdots+\phi^h)}$</td></tr>
</table>
After choosing trend (diag in lect 6 notes), need include seasonal component and error term, each either additively or multiplicatively.
For exponential smoothing mtds, type of error don't matter – point forecasts will be the same. Errors only affect PI of ETS models
Ignoring the error components, there are 15 basic exponential smoothing mtds/ ETS model (Error, Trend, Seasonal) to consider
<table>
<tr><td>Trend</td><td>Seasonal None</td><td>Seasonal Add</td><td>Seasonal Mult</td></tr>
<tr><td>N (None)</td><td>N,N (SES)</td><td>N,A</td><td>N,M</td></tr>
<tr><td>A (Additive)</td><td>A,N (Holt's linear mtd)</td><td>A,A (Additive Holt-Winters mtd)</td><td>A,M (Mult Holt-Winters mtd)</td></tr>
<tr><td>A_d (Additive damped)</td><td>A_d,N (Additive damped trend mtd)</td><td>A_d,A</td><td>A_d,M</td></tr>
<tr><td>M (Mult)</td><td>M,N (Exp trend mtd)</td><td>M,A</td><td>M,M</td></tr>
<tr><td>M_d (Mult damped)</td><td>M_d,N (Mult Holt-Winters mtd)</td><td>M_d,A</td><td>M_d,M</td></tr>
</table>
For each model, there are 2 possible state space model. If same params are used, both models give same point forecasts, but diff PI</td></tr>
<tr><td>SES (N,N) – Simple exponential smoothing</td><td>Definition 1) $\hat{y}_{t+1} = \hat{y}_t + \alpha(y_t - \hat{y}_t)$ = adjusting forecast for next period using forecast error from previous period, where $\alpha \in [0,1]$
Defn 2) $\hat{y}_{t+1} = \alpha y_t + (1-\alpha)\hat{y}_t$ = weighted avg of most recent obs and most recent forecast
$\hat{y}_{t+1} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \cdots + (1-\alpha)^t \hat{y}_1$. Thus the forecast is a weighted avg of all past obs
For this defn, we need to know values of α and $\hat{y}_1$. Size of α determines impact of $\hat{y}_1$ on forecast
Deciding on value of $\hat{y}_1$ to use = initialisation problem
Defn 3) $l_t = \alpha y_t + (1-\alpha)l_{t-1}$. $\hat{y}_{t+1} = l_t$. Via a) evolution of components (only consider level of series, l_t) and b) forecast fn.
a) can be considered as the transition eqn and b) as the measurement eqn
If we want to forecast for a longer forecast horizon, SES model returns a fixed value.
Thus forecasts eqn should have been $\hat{y}_{t+h|t} = \hat{y}_{t+1|t} = l_t$. Only suitable for series w no trend or seasonal component</td></tr>
<tr><td></td><td>For defn 2: can show that sum of weights = 1. $\hat{y}_{t+1} = \sum_{k=0}^{t-1}\alpha(1-\alpha)^k y_t + (1-\alpha)^t \hat{y}_1$
So $\sum_{k=0}^{t-1}\alpha(1-\alpha)^k + (1-\alpha)^t = \frac{\alpha(1-(1-\alpha)^t)}{1-(1-\alpha)} + (1-\alpha)^t = 1 - (1-\alpha)^t + (1-\alpha)^t = 1$. Note $\sum_{i=1}^{n}\alpha r^i = \frac{\alpha(1-r^n)}{1-r}$</td></tr>
<tr><td>Holt's Linear Mtd (A,N)</td><td>Assume that at each point, there is a linear trend b_t and level l_t from which that trend starts. Need to estimate $\alpha, \beta^*, l_0, b_0$
2 smoothing eqns for each component. l_t and b_t. Forecast eqn: $\hat{y}_{t+h|t} = l_t + hb_t$ (linear fn of h)</td></tr>
</table>

	l_t is an estimate of level at time t = weighted avg of y_t and the one-step-ahead forecast $(l_{t-1}+b_{t-1})$ b_t is an estimate of trend at time t = weighted avg of b_{t-1} and the current tred, estimated by $l_t - l_{t-1}$ α = smoothing param for the level, while β^* = smoothing param for the trend. Both α and $\beta^* \in [0,1]$ Holt's linear mtd assume data follow a constant trend indefinitely into future. Empirical evidence shot mtd tend to over-forecast
Additive Damped Trend (A_d,N)	Damped trend mtd introduces a param that weakens/softens the trend to a flat line some time in the future Effect of trend is damped each time it enters the forecast and level fns. And $\hat{y}_{t+h\|t} = l_t + (\phi + \phi^2 + \cdots + \phi^h)b_t$ Since $\phi \in [0,1]$, $\lim_{h\to\infty} \hat{y}_{t+h\|t} = l_t + \frac{\phi}{1-\phi} b_t$. This means in short term, forecasts have a trend but in long run, they are constant When ϕ = 1, model = Holt's linear mtd. ϕ rarely set to be < 0.8, since it has a strong effect. Usually $\phi \in [0.8, 0.98]$
Holt-Winters Seasonal Mtds	Holt-Winters seasonal models contains 3 components: trend b_t, level l_t, and the seasonal component s_t. Corresponding smoothing params = α, β^*, γ. Freq of seasonality = m (for quartely data, m = 4) If seasonal variations roughly constant in series: Use Additive mtd. In this model, seasonal component add up to ≈ 0 within each year If seasonal variations prop to level of series: Use Mult mtd. In this model, seasonal components add up to ≈ m within each year
	Additive seasonality (A,A). $\hat{y}_{t+h\|t} = l_t + hb_t + s_{t-m+h_m^*}$, where $h_m^* = \lfloor (h-1) \bmod m \rfloor + 1$ Level eqn = weighted avg btw seasonally adjusted obs $(y_t - s_{t-m})$ and the non-seasonal forecast $(l_{t-1}+b_{t-1})$ Seasonal eqn = weighted avg btw current seasonal index $(y_t - l_{t-1} - b_{t-1})$ and seasonal index of the same season in the previous year Equivalent formulation for seasonal component is $s_t = \gamma^*(y_t - l_t) + (1-\gamma^*)s_{t-m}$, where $\gamma = \gamma^*(1-\alpha)$ "Proof": Using level eqn, $s_t = \gamma^*(y_t - l_t) + (1-\gamma^*)s_{t-m} = \gamma^*[y_t - \alpha(y_t - s_{t-m}) - (1-\alpha)(l_{t-1}+b_{t-1})] + (1-\gamma^*)s_{t-m} =$ $\gamma^*[(1-\alpha)(y_t - l_{t-1} - b_{t-1}) + \alpha s_{t-m}] + (1-\gamma^*)s_{t-m} = \gamma^*(1-\alpha)(y_t - l_{t-1} - b_{t-1}) + (\gamma^*\alpha + 1 - \gamma^*)s_{t-m} = \gamma(y_t - l_{t-1} -$ $b_{t-1}) + (1-(1-\alpha)\gamma^*)s_{t-m} = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$
	Mult Seasonality (A,M). $\hat{y}_{t+h\|t} = (l_t + hb_t)s_{t-m+h_m^*}$, where $h_m^* = \lfloor (h-1) \bmod m \rfloor + 1$

Summary

Note h-1 = km + $\lfloor (h-1) \bmod m \rfloor$, where k ∈ ℤ. So $t + h - m(k+1) = t - m + h_m^*$ (i.e. h - km = h_m^*). Let $\phi_h = \phi + \phi^2 + \cdots + \phi^h$

Season / Trend	N	A	M
N	$\hat{y}_{t+h\|t} = l_t$ $l_t = \alpha y_t + (1-\alpha)l_{t-1}$	$\hat{y}_{t+h\|t} = l_t + s_{t-m+h_m^*}$ $l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)l_{t-1}$ $s_t = \gamma(y_t - l_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = l_t s_{t-m+h_m^*}$ $l_t = \alpha(y_t/s_{t-m}) + (1-\alpha)l_{t-1}$ $s_t = \gamma(y_t/l_{t-1}) + (1-\gamma)s_{t-m}$
A	$\hat{y}_{t+h\|t} = l_t + hb_t$ $l_t = \alpha y_t + (1-\alpha)(l_{t-1}+b_{t-1})$ $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = l_t + hb_t + s_{t-m+h_m^*}$ $l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1}+b_{t-1})$ $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (l_t + hb_t)s_{t-m+h_m^*}$ $l_t = \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1}+b_{t-1})$ $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma\frac{y_t}{l_{t-1}+b_{t-1}} + (1-\gamma)s_{t-m}$
A_d	$\hat{y}_{t+h\|t} = l_t + \phi_h b_t$ $l_t = \alpha y_t + (1-\alpha)(l_{t-1}+\phi b_{t-1})$ $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)\phi b_{t-1}$	$\hat{y}_{t+h\|t} = l_t + \phi_h b_t + s_{t-m+h_m^*}$ $l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1}+\phi b_{t-1})$ $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - l_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (l_t + \phi_h b_t)s_{t-m+h_m^*}$ $l_t = \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1}+\phi b_{t-1})$ $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma\frac{y_t}{l_{t-1}+\phi b_{t-1}} + (1-\gamma)s_{t-m}$
M	$\hat{y}_{t+h\|t} = l_t b_t^h$ $l_t = \alpha y_t + (1-\alpha)l_{t-1}b_{t-1}$ $b_t = \beta^*(l_t/l_{t-1}) + (1-\beta^*)b_{t-1}$		

State Space Models

For each mtd above, there are 2 state space models - 1 w additive errors, 1 w multiplicative errors
For state space model, have to identify 1) state vector and 2) source of error that appears in both state and measurement eqn

ADDITIVE ERROR MODELS

Trend	Seasonal N	Seasonal A	Seasonal M
N	$y_t = \ell_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$	$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/\ell_{t-1}$
A	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + b_{t-1})$
A_d	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = \phi b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + \phi b_{t-1})$

ETS(A,M,N):
$y_t = l_{t-1}b_{t-1} + e_t$
$l_t = l_{t-1}b_{t-1} + \alpha e_t$
$b_t = b_{t-1} + \beta e_t/l_{t-1}$

MULTIPLICATIVE ERROR MODELS

Trend	Seasonal N	Seasonal A	Seasonal M
N	$y_t = \ell_{t-1}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$	$y_t = (\ell_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \alpha(\ell_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + s_{t-m})\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
A	$y_t = (\ell_{t-1} + b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
A_d	$y_t = (\ell_{t-1} + \phi b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$

ETS(A,A,N). Want it in the form of $y_t = w^T x_{t-1} + e_t$ and $x_t = Fx_{t-1} + ge_t$. And $x_t = \begin{bmatrix} l_t \\ b_t \end{bmatrix}$

	We know $e_t = y_t - \hat{y}_t$. And for (A,A,N), $y_t = \hat{y}_t + e_t = l_{t-1} + b_{t-1} + e_t = [1 \quad 1]x_{t-1} + e_t$ (for 1-step ahead forecast) $l_t = \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1}) = l_{t-1} + b_{t-1} + \alpha(y_t - l_{t-1} - b_{t-1}) = l_{t-1} + b_{t-1} + \alpha(y_t - \hat{y}_t) = [1 \quad 1]x_{t-1} + \alpha e_t$ $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} = b_{t-1} + \beta^*(l_t - l_{t-1} - b_{t-1}) = b_{t-1} + \beta^* \alpha e_t = [0 \quad 1]x_{t-1} + \beta e_t$, where $\beta = \beta^* \alpha$ So $y_t = [1 \quad 1]x_{t-1} + e_t$. $x_t = \begin{bmatrix} l_t \\ b_t \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_{t-1} + \begin{bmatrix} \alpha \\ \beta \end{bmatrix} e_t$. Since $\mu_{t+h\mid t} = E(y_{t+h} \mid x_t)$ Suppose $e_t \sim GWN(0, \sigma^2)$, then $\mu_{t\mid t-1} = E(y_t \mid x_{t-1}) = E([1 \quad 1]x_{t-1} + e_t) = l_{t-1} + b_{t-1}$ = forecast from Holt's linear mtd For most models, $\mu_{t+h\mid t} = \hat{y}_{t+h\mid t}$. But will not hold for models w multiplicative trend or multiplicative seasonality for h ≥ 2
	ETS(M,A,N). $l_t = \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1})$ and $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$ and $\hat{y}_{t+h\mid t} = l_t + hb_t$. $\hat{y}_t = l_{t-1} + b_{t-1}$ However, now relative error = $e_t = \frac{y_t - \hat{y}_t}{\hat{y}_t}$. So $y_t = e_t \hat{y}_t + \hat{y}_t = \hat{y}_t(1 + e_t) = (l_{t-1} + b_{t-1})(1 + e_t)$ And $x_t = \begin{bmatrix} l_t \\ b_t \end{bmatrix}$ $l_t = l_{t-1} + b_{t-1} + \alpha(y_t - \hat{y}_t) = (l_{t-1} + b_{t-1})\left[1 + \frac{\alpha(y_t - \hat{y}_t)}{\hat{y}_t}\right] = (l_{t-1} + b_{t-1})[1 + \alpha e_t]$. So $l_t - l_{t-1} - b_{t-1} = \alpha e_t(l_{t-1} + b_{t-1})$ $b_t = \beta^*(l_t - l_{t-1} - b_{t-1}) + b_{t-1} = \beta^*(\alpha e_t(l_{t-1} + b_{t-1})) + b_{t-1} = b_{t-1} + \beta e_t(l_{t-1} + b_{t-1})$, where $\beta = \beta^* \alpha$ So $y_t = [1 \quad 1]x_{t-1}(1 + e_t)$. $x_t = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_{t-1} + [1 \quad 1]x_{t-1} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} e_t$ Suppose $e_t \sim GWN(0, \sigma^2)$, then $\mu_{t\mid t-1} = E(y_t \mid x_{t-1}) = E([1 \quad 1]x_{t-1}(1 + e_t)) = l_{t-1} + b_{t-1}$ = forecast from Holt's linear mtd
	In general, $y_t = w(x_{t-1}) + r(x_{t-1})e_t$ and $x_t = f(x_{t-1}) + g(x_{t-1})e_t$. Full model list in lect 6 For additive models, $r(x_{t-1}) = 1$, $e_t = y_t - \hat{y}_t$. For multiplicative models, $e_t = \frac{y_t - \hat{y}_t}{\hat{y}_t}$ The models with multiplicative error/trend/seasonality could involve division by 0, and so are numerically unstable. The multiplicative error models are unstable when the data values contain zeros or negative values. If the data are not strictly positive, use only the 6 fully additive models.
Computations	With the state space models, and given x_0, y_1. We can compute $\hat{y}_1, e_1, x_1, \ldots$
Residuals & Forecast errors	For *forecast* package, 1-step forecasts defined as $y_t - \hat{y}_t$. The residuals are the estimates of the innovation (forecast) errors. For the state space models with additive errors, residuals = one-step forecast (innovation) errors. For models with multiplicative errors, residuals = $y_t - \hat{y}_t$. Forecast/innovations = $\frac{y_t - \hat{y}_t}{\hat{y}_t}$
Linear Innovations State Space models	$y_t = w'x_{t-1} + e_t$ – (measurement eqn), $w'x_{t-1}$ describes effect of past on y_t. And $x_t = Fx_{t-1} + ge_t$ – (transition eqn) y_t denotes observed values, x_t is state vector containing info on level, growth and seasonal patterns. Error term, $e_t \sim$ GWN(0, σ^2) and is the only source of noise in model; also known as the innovation in the model F is transition matrix. Fx_{t-1} describes effect of past on current state x_t. Vector g determines extent of effect of innovation on state Vectors w, g and matrix F is fixed over time (in this course). And are parameters we need to estimate Given the initial state vector, the pdf for **y** = $p(\boldsymbol{y} \mid x_0) = p(y_1, \ldots, y_n \mid x_0) = p(y_n \mid y_1, \ldots, y_{n-1}, x_0) \times p(y_{n-1} \mid y_1, \ldots, y_{n-2}, x_0) \times \ldots \times p(y_1 \mid x_0) = p(y_n \mid x_{n-1}) p(y_{n-1} \mid x_{n-2}) \ldots p(y_1 \mid x_0) = \prod_{t=1}^{n} p(y_t \mid x_{t-1}) = \prod_{t=1}^{n} p(e_t)$ Since assume e_t are Gaussian, $p(\boldsymbol{y} \mid x_0) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n exp\left(-\frac{1}{2\sigma^2}\sum_{t=1}^{n} e_t^2\right)$
	ETS(A,N,N). $y_t = l_{t-1} + e_t$. $l_t = l_{t-1} + \alpha e_t$. Where $x_t = l_t, w = 1, F = 1, g = \alpha$ When $\alpha = 0$, local levels l_t dont change at all: $y_t = l + e_t$ When $\alpha = 0$, model is a random walk model: $y_t = y_{t-1} + e_t$ Conditional expectation for 1-step forecast is $\mu_{t+1} = E(y_{t+1} \mid x_t) = E(l_t + e_{t+1} \mid x_t) = l_t = l_{t-1} + \alpha e_t = l_{t-1} + \alpha(y_t - l_{t-1}) = (1 - \alpha)l_{t-1} + \alpha y_t = \cdots = (1-\alpha)^t l_0 + \alpha \sum_{j=0}^{t-1} (1-\alpha)^j y_{t-j}$ Conditional var $Var(y_t \mid x_{t-1}) = Var(l_t + e_{t+1} \mid x_{t-1}) = \sigma^2$ If 0 < $(1-\alpha)$ < 1, then forecast can be interpreted as a weighted avg of previous values, w older values being assigned less weight. For this model, stability condition is satisfied if 0 < α < 2 Note that $y_t = l_0 + e_t + \alpha \sum_{j=1}^{t-1} e_j$
	ETS(A,A,N). $y_t = l_{t-1} + b_{t-1} + e_t$. $l_t = l_{t-1} + b_{t-1} + \alpha e_t$. $b_t = b_{t-1} + \beta e_t$ Where $x_t = \begin{bmatrix} l_t \\ b_t \end{bmatrix}, w = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, F = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, g = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ When $\alpha > 0, \beta > 0$ and $2\alpha + \beta < 4$, then model is stable However, in practice, restrictions 0 < α < 1 and 0 < β < α are usually applied. Corresponding exponential smoothing mtd is (A,N).
	ETS(A,A,A). $y_t = l_{t-1} + b_{t-1} + s_{t-m} + e_t$. $l_t = l_{t-1} + b_{t-1} + \alpha e_t$. $b_t = b_{t-1} + \beta e_t$. $s_t = s_{t-m} + \gamma e_t$ $x_t = \begin{bmatrix} l_t \\ b_t \\ s_t \\ s_{t-1} \\ \vdots \\ s_{t-m+1} \end{bmatrix}, w = \begin{bmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, F = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 1 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, g = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ Seasonal components are normalised to prevent confounding w the level. The usual parameter space are 0 < α < 1 and 0 < β < α and 0 < γ < 1 - α
Nonlinear innovations State Space Models	$y_t = w(x_{t-1}) + r(x_{t-1})e_t$ and $x_t = f(x_{t-1}) + g(x_{t-1})e_t$, where functions w and r take in vector and return scalar, while fns f and g returns a vector, e_t is a WN process. Joint dist of variables: $p(\boldsymbol{y} \mid x_0) = \prod_{t=1}^{n} p(y_t \mid x_{t-1}) = \prod_{t=1}^{n} p(e_t) / \lvert r(x_{t-1}) \rvert$ Assume e_t follow Gaussian dist, $p(\boldsymbol{y} \mid x_0) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \lvert \prod_{t=1}^{n} r(x_{t-1}) \rvert^{-1} exp\left(-\frac{1}{2\sigma^2}\sum_{t=1}^{n} e_t^2\right)$
	ETS(M,N,N). $y_t = l_{t-1}(1 + e_t)$. $l_t = l_{t-1}(1 + \alpha e_t)$. So $y_t = l_{t-2}(1 + \alpha e_{t-1})(1 + e_t) = \cdots = l_0(1 + e_t)\prod_{j=1}^{t-1}(1 + \alpha e_j)$ Where state vector $x_t = l_t$. $w(x_{t-1}) = r(x_{t-1}) = f(x_{t-1}) = l_{t-1}$. $g(x_{t-1}) = \alpha l_{t-1}$ $\mu_{t\mid t-1} = E(y_t \mid x_{t-1}) = E(l_{t-1} + l_{t-1}e_t \mid x_t) = l_{t-1} + l_{t-1}E[e_t \mid x_t] = l_{t-1} + l_{t-1}(0) = l_{t-1} = \hat{y}_t$ (same as ETS(A,N,N)) But conditional var $Var(y_t \mid x_{t-1}) = Var(l_{t-1} + l_{t-1}e_t \mid x_t) = Var(l_{t-1}e_t \mid x_t) = l_{t-1}^2 \sigma^2$ (diff from ETS(A,N,N)) So forecast var will depend on level of process (leading to diff PI) When $\alpha = 0$, state does not change; i.e. identical to additive model except for a parametrisation When $\alpha = 1$, model is $y_t = y_{t-1}(1 + e_t)$
	ETS(M,A,N). $y_t = (l_{t-1} + b_{t-1})(1 + e_t)$. $l_t = (l_{t-1} + b_{t-1})(1 + \alpha e_t)$. $b_t = b_{t-1} + \beta(l_{t-1} + b_{t-1})e_t$ Where $x_t = \begin{bmatrix} l_t \\ b_t \end{bmatrix}, w(x_{t-1}) = r(x_{t-1}) = l_{t-1} + b_{t-1}$, $f(x_{t-1}) = \begin{bmatrix} l_{t-1} + b_{t-1} \\ b_{t-1} \end{bmatrix}, g = \begin{bmatrix} \alpha(l_{t-1} + b_{t-1}) \\ \beta(l_{t-1} + b_{t-1}) \end{bmatrix}$ Special cases: $\beta = 0 \equiv$ global trend. $\beta = 0, \alpha = 1 \equiv$ random walk w drift. $\beta = 0, \alpha = 0 \equiv$ fixed level and trend
	Initial state x_0, and params are unknown and have to be estimated from data - Smoothing params, e.g. α and β for ETS(A,A,N) model. Refer to these as a vector θ

Estimation in State Space models

- Initial state x_0. - Innovations var σ^2

MLE: Likelihood fn for generatl state space model is $p(y|\theta, x_0, \sigma^2) = \prod_{t=1}^{n} \frac{p(e_t)}{|r(x_{t-1})|}$

Assuming Gaussian innovations, likelihood can be written as $L(\theta, x_0, \sigma^2) = (2\pi\sigma^2)^{-n/2} |\prod_{t=1}^{n} r(x_{t-1})|^{-1} exp\left(-\frac{1}{2\sigma^2}\sum_{t=1}^{n} e_t^2\right)$

Log-likelihood, $l(\theta, x_0, \sigma^2) = -\frac{n}{2} log(2\pi\sigma^2) - \sum_{t=1}^{n} log|r(x_{t-1})| - \frac{1}{2\sigma^2}\sum_{t=1}^{n} e_t^2$

To maximise log-likelihood: $\frac{\partial l}{\partial \sigma^2} = -\frac{n}{2}\frac{2\pi}{2\pi\sigma^2} - \frac{1}{2}(-1)(\sigma^2)^{-2}\sum_{t=1}^{n} e_t^2 = 0$ to get $\hat{\sigma}^2 = \frac{1}{n}\sum_{t=1}^{n} e_t^2$

Then log-likelihood becomes $l(\theta, x_0, \sigma^2) = -\frac{n}{2} log(2\pi\hat{\sigma}^2) - \sum_{t=1}^{n} log|r(x_{t-1})| - \frac{1}{2\hat{\sigma}^2}\sum_{t=1}^{n} e_t^2 = -\frac{n}{2} log(\hat{\sigma}^2) - \frac{n}{2} log(2\pi) - \sum_{t=1}^{n} log|r(x_{t-1})| - \frac{n}{2} = -\frac{n}{2} log(\hat{\sigma}^2) - \frac{n}{2} log(2\pi e) - \sum_{t=1}^{n} log|r(x_{t-1})|$

Equivalent to min $S(\theta, x_0) = |\prod_{t=1}^{n} r(x_{t-1})|^{2/n} \sum_{t=1}^{n} e_t^2$ aka augmented sum of squares criterion (power to 2/n L(), ignore constant?)
Instead of using the likelihood function, we could target to find the parameters θ, x_0 by minimising the one-step MSE, MAE or some other error metric. Yet another method is to minimise the residual variance.

Num of parameters. Suppose we have weekly data, i.e. m = 52. And we wish to fit an ETS(A,A,A) model, we would then have to estimate 52 + 2 = 54 seed states (l_0, b_0) and 3 smoothing parameters (α, β, γ)
Huge num = hard to compute. Solution: use heuristic methods of estimation OR assume certain weeks have same effects

Initial values of x_0. 1) For initial seasonal component, perform a classical decomposition of the process
2) For the initial level component, perform a linear regression of the first y_t values on 1,2,…,10 and use intercept term as l_0
3) For initial growh component, use slop estimated from 2) as b_0 if it is additive trend. If multiplicative trend, use $b_0 = 1 + b/a$, where b is slope from 2) and a is intercept

Prediction Intervals

When forecasting TS, sources of uncertainty are 1) model choice, 2) future innovations e_{t+1}, e_{t+2}, …, 3) Parameters estimates
In practice, we only consider 2) the uncertainty in future innovations
Prediction dist = dist of future values, given the model, its estimated parameters and x_t. So $y_{t+h|t} \equiv y_{t+h|x_t}$
Forecast mean is $\mu_{t+h|t} = E(y_{t+h}|x_t)$. Forecast variance is $v_{t+h|t} = Var(y_{t+h}|x_t)$.

Analytical expressions for the forecast var are only available for some of the models
Class 1: easy to derive
Class 2 and 3: can derive, but involve making a few further assumptions
Class 4 and 5: cannot derive, use simulation to obtain prediction intervals

Class 1 →	A,N,N	A,N,A			
	A,A,N	A,A,A			
	A,A_d,N	A,A_d,A			
	M,N,N	M,N,A	M,N,M		
Class 2 →	M,A,N	M,A,A	M,A,M	← Class 3	
	M,A_d,N	M,A_d,A	M,A_d,M		
Class 4 →	M,M,N		M,M,M		
	M,M_d,N		M,M_d,M		
		M,M,A	A,N,M	A,M,N	A,M_d,N
Class 5 →		M,M_d,A	A,A,M	A,M,A	A,M_d,A
			A,A_d,M	A,M,M	A,M_d,M

ETS(A,N,N). $y_t = x_{t-1} + e_t$. $x_t = x_{t-1} + \alpha e_t$. Where $x_t = l_t, w = 1, F = 1, g = \alpha$
One-step conditional mean is $\mu_{t+1} = E(y_{t+1}|x_t) = E(x_t + e_{t+1}|x_t) = l_t$
Prediction error, or var of this forecast = $Var(y_{t+1}|x_t) = Var(x_t + e_{t+1}|x_t) = \sigma^2$
When h = 2, $E(y_{t+2}|x_t) = E(x_{t+1} + e_{t+2}|x_t) = E(x_{t+1}|x_t) = E(x_t + \alpha e_{t+1}|x_t) = x_t$
Prediction error, or var of this forecast = $Var(y_{t+2}|x_t) = Var(x_{t+1} + e_{t+2}|x_t) = \sigma^2 + Var(x_{t+1}|x_t) = \sigma^2 + Var(x_t + \alpha e_{t+1}|x_t) = \sigma^2 + \alpha^2\sigma^2 = \sigma^2(1+\alpha^2)$
In general, $\hat{y}_{t+h|t} = w'F^{h-1}x_t = l_t$. And $v_{t+h|t} = var(y_{t+h|t}|x_t) = \cdots = \alpha^2(h-1)\sigma^2 + \sigma^2$

In general for class 1 models, $y_t = w'x_{t-1} + e_t$. $x_t = Fx_{t-1} + ge_t$

E.g. for ETS(A,N,A), $x_t = \begin{bmatrix} l_t \\ s_t \\ s_{t-1} \\ \vdots \\ s_{t-(m-1)} \end{bmatrix}, w = \begin{bmatrix} 1 \\ 0_{m-1} \\ 1 \end{bmatrix}, F = \begin{bmatrix} 1 & 0'_{m-1} & 0 \\ 0 & 0'_{m-1} & 1 \\ 0_{m-1} & I_{m-1} & 0_{m-1} \end{bmatrix}, g = \begin{bmatrix} \alpha \\ \beta \\ 0_{m-1} \end{bmatrix}$

Then $\hat{y}_{t+h|t} = E(y_{t+h}|x_t) = w'E(x_{t+h-1}|x_t) = w'E(Fx_{t+h-2} + ge_{t+h-1}|x_t) = w'FE(x_{t+h-2}|x_t) = \cdots = w'F^{h-1}x_t$

Forecast var = $v_{t+h|t} = w'Var(x_{t+h-1}|x_t)w + \sigma^2$. Which can be simplified to $v_{T+h|T} = \begin{cases} \sigma^2 & if\ h = 1 \\ \sigma^2[1 + \sum_{j=1}^{h-1} c_j^2] & if\ h \geq 2 \end{cases}$ $(c_j = w'F^{j-1}g)$

Model	Forecast var σ_h^2	$c_j = w'F^{j-1}g$
(A,N,N)	$\sigma_h^2 = \sigma^2[1 + \alpha^2(h-1)]$	α
(A,A,N)	$\sigma_h^2 = \sigma^2[1 + (h-1)\{\alpha^2 + \alpha\beta h + \frac{1}{6}\beta^2 h(2h-1)\}]$	$\alpha + \beta j$. $F^h = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix}$
(A,A_d,N)	$\sigma_h^2 = \sigma^2\left[1 + \alpha^2(h-1) + \frac{\beta\phi h}{(1-\phi)^2}\{2\alpha(1-\phi) + \beta\phi\} - \frac{\beta\phi(1-\phi^h)}{(1-\phi)^2(1-\phi^2)}\{2\alpha(1-\phi^2) + \beta\phi(1 + 2\phi - \phi^h)\}\right]$	
(A,N,A)	$\sigma_h^2 = \sigma^2[1 + \alpha^2(h-1) + \gamma k(2\alpha + \gamma)]$	
(A,A,A)	$\sigma_h^2 = \sigma^2\left[1 + (h-1)\{\alpha^2 + \alpha\beta h + \frac{1}{6}\beta^2 h(2h-1)\} + \gamma k\{2\alpha + \gamma + \beta m(k+1)\}\right]$	
(A,A_d,A)	$\sigma_h^2 = \sigma^2\left[1 + \alpha^2(h-1) + \gamma k(2\alpha + \gamma) + \frac{\beta\phi h}{(1-\phi)^2}\{2\alpha(1-\phi) + \beta\phi\} - \frac{\beta\phi(1-\phi^h)}{(1-\phi)^2(1-\phi^2)}\{2\alpha(1-\phi^2) + \beta\phi(1 + 2\phi - \phi^h)\} + \frac{\beta\gamma\phi}{(1-\phi)(1-\phi^m)}\{k(1-\phi^m) - \phi^m(1-\phi^{mk})\}\right]$	

Intervals via Simulation. Suppose required forecast horizon is h w model conditional on most recent state x_T. Then for i = 1,…,M,
1) Generate obs $y_{T+1}^i, y_{T+2}^i, \dots, y_{T+h}^i$, starting w x_T, from the fitted model
2a) Each e_{T+k} valus is obtained from a random num generator assuming a Gaussian or other appropriate dist OR
2b) Bootstrap to resample from historical values of e_t if unsure about innovations dist (bootstrap also appropriate when e_t are Gaussian but y_t are not, due to it being a non-linear model)
Usually take M = 5000. Then take mean of simulated values at each h as the point forecast. E.g. h = 1, take mean of $\{y_{T+1}^1, y_{T+1}^2, \dots, y_{T+1}^M\}$
Can use quantiles to obtain PI. E.g. for 95% PI for h = 3, take 0.025 and .975 quantiles of $\{y_{T+3}^1, y_{T+3}^2, \dots, y_{T+3}^M\}$
mtd4 <- model(TS, add = ETS(Y ~ error("A") + trend(method="A") + season("A")))
mtd4_fc <- forecast(mtd4, h=16); autoplot(mtd4_fc, data=TS, level=95) + labs(title = "Prediction intervals")

Model Selection

1. Split the time series into a training and test set
2. Fit each model using the training set (via MLE).

	3. Assess the forecast accuracy of each model using the test set. 4. Choose the model with the lowest forecast accuracy. 5. Refit the chosen model to the full time series and use these new parameters for forecasting future observations. 6. An alternative was to use cross-validation to select the best model.
	Sometimes, test set is too small to draw reliable conclusions or diff to decide which error metric to use Instead, can use penalized likelihood method. This fits model to entire data, and compute likelihood for that data Model w highest likelihood is chosen. However, likelihood is penalised for num of params used. $AIC = -L(\theta, x_0) + 2q$, where q = num of params in θ + num of free states in x_0 OR $BIC = -L(\theta, x_0) + qlog(T)$
	mtd5 <- model(TS, ets1 = ETS(Y)) # will fit best ETS model. *report(mtd5); gg_tsresiduals(mtd5)*
Tut 7	Theta Decomposition
	stl_seas_adj <- model(ts, stl_robust2 = STL(Y ~ season(window=5), robust=TRUE)) %>% components() %>% as_tsibble() *ses1 <- select(stl_seas_adj, season_adjust) %>% model(ses_model1 = ETS(season_adjust ~ error("A") + trend("N") + season("N")))* *report(ses1)* # For SES model, can see α and $\hat{y}_1 = \mathrm{l}_0$ w optimal α (RMSE) *ses2 <- select(stl_seas_adj, season_adjust) %>% model(opt_alpha = ETS(season_adjust ~ error("A") + trend("N") + season("N")), fixed_alpha = ETS(season_adjust ~ error("A") + trend("N", alpha=0.1) + season("N")))* *autoplot(augment(ses2), .vars= season_adjust, col="gray") + geom_line(aes(y=.fitted, col=.model))* *accuracy(ses2)*

Stationarity	A strictly stationary TS is one for which the joint dist of $\{y_{t_1}, y_{t_2}, \dots, y_{t_k}\}$ is identical to the dist of $\{y_{t_1+h}, y_{t_2+h}, \dots, y_{t_k+h}\}$ for all k, all time points $t_1, t_2, \dots, t_k$ and all h = 0, ±1, ±2, … Strict stationarity implies $E(y_s) = E(y_t) = \mu$ for all s,t. And ACVF = $\gamma_{s,t} = cov(y_s, y_t) = \gamma_{s+h,\ t+h}$ for all s,t and h
	A weakly stationary TS is a finite variance process s.t. 1) μ_t is a constant (don't depend on t) & 2) $\gamma_{s,t}$ depends on s and t only through s - t For this course, stationary = weakly stationary. Let μ denote mean and h = s - t and ACVF = $\gamma_{s,t} = \gamma_{s-t,0} = \gamma_h$
	E.g. 3-MA filter on WN process $e_t \sim WN(0, \sigma^2)$. $y_t = \frac{e_{t-1}+e_t+e_{t+1}}{3}$ ACVF = $\gamma_{s,t} = \begin{cases} 3\sigma^2/9 & s = t \\ 2\sigma^2/9 & \|s-t\| = 1 \\ \sigma^2/9 & \|s-t\| = 2 \\ 0 & \|s-t\| \geq 3 \end{cases}$. Since $E(y_t) = 0$ = constant, and ACVF depends on s-t, y_t is stationary E.g. 2: $y_t = \beta_0 + \beta_1 t + e_t$. $E(y_t) = \beta_0 + \beta_1 t$ which is not indep of t, so y_t is not stationary E.g. 3: $y_t = y_{t-1} + e_t$. $y_0 = 0$. $Cov(y_1, y_2) = cov(y_0 + e_1, y_1 + e_2) = cov(y_0 + e_1, y_0 + e_1 + e_2) = \sigma^2$ But $cov(y_2, y_3) = cov(y_0 + e_1 + e_2, y_0 + e_1 + e_2 + e_3) = 2\sigma^2$. So even though s - t is the same, ACVF is diff, so y_t is not stationary
	Properties of stationary processes: 1) $\gamma_0 = var(y_t)$. 2) $\|\gamma_h\| \leq \gamma_0$ (can be proved using Cauchy-Schwarz inequality). 3) $\gamma_h = \gamma_{-h}$ If $\{y_t\}$ is a stationary TS, then for all s, the joint dist of $(y_t, \dots, y_{t+s})$ don't depend on t A stationary series is - roughly horizontal, - has constant var, - has no predictable patterns in long-term, - a series w trend and/or seasonality is not stationary due to its changing mean
	For a stationary series: - ACF drops to 0 relatively quickly For non stationary data: - ACF decreases slowly, value of r is often large and positive for many lags To have stationary TS, can use transformations to stabilize var OR to difference the data
Differencing	A single differencing stabilizes the mean of a TS by removing changes in the level of a TS. $y'_t = y_t - y_{t-1}$ The differenced series will have only T-1 values. *mutate(df, diff1 = difference(Y)) %>% gg_tsdisplay(diff1, plot_type = 'histogram')* *mutate(df, diff1 = difference(Y)) %>% features(diff1, feature_set(tags='portmanteau')*
	Occasionally, differenced data will not appear stationary, and may be necessary to difference data a second time. $y''_t = y'_t - y'_{t-1} = y_t - y_{t-1} - (y_{t-1} - y_{t-2}) = y_t - 2y_{t-1} + y_{t-2}$. In practice, shouldn't be neccesary to go beyond 2nd order difference
	Seasonal differencing. $y'_t = y_t - y_{t-m}$, where m = num of seasons. For monthly data, m = 12 The new series = "lag-m differences". If seasonal differenced data appears to be WN, then an appropriate model would be $y_t = y_{t-m} + e_t$ = seasonal naive
	Twice differenced series. E.g do both seasonal and a first difference $y''_t = y'_t - y'_{t-1} = y_t - y_{t-m} - (y_{t-1} - y_{t-m-1}) = y_t - y_{t-1} - y_{t-m} + y_{t-m-1}$ When both seasonal and first diff are applied, doesn't matter which is done first However, if there is a strong seasonal pattern, seasonal differencing should be done first. This is because, sometimes, the seasonally differenced series alone is close enough to stationarity - there might be no need to do the first differencing as well. If we performed the first differencing first, the strong seasonality would compel us to perform the second (seasonal) differencing too. First differences are the change from one obs and the next. Seasonal differences are the change between one year and the next. Higher order differencing should be avoided as they are difficult to intrepret.
	Unit Root Tests. The Kwaitkowski-Phillips-Schmidt-Shin (KPSS) test can be used to test if a series is stationary H_0: data is stationary and non-seasonal. H_1: data is not stationary KPSS test can be repeatedly applied to successive differencing to determine num of differencings that should be carried out. *features(TS, Y, list(unitroot_ndiffs)).* # Use *unitroot_nsdiffs* to determine optimal num of seasonal differencing
	Backshift Notation. $By_t = y_{t-1}$. i.e. shift data back one period. $B(By_t) = B^2y_t = y_{t-2}$. Note Bc = c (where c is a constant) For monthly data, to denote same month last year, $B^{12}y_t = y_{t-12}$ First difference: $y'_t = y_t - y_{t-1} = y_t - By_t = (1 - B)y_t$. Second order diff: $y''_t = y_t - 2y_{t-1} + y_{t-2} = (1 - B)^2 y_t$ In general, a d^{th} order diff can be written as $(1 - B)^d y_t$ Seasonal diff followed by a first diff: $(1 - B)(1 - B^m)y_t = (1 - B - B^m + B^{m+1})y_t = y_t - y_{t-1} - y_{t-m} + y_{t-m-1}$
Auto-regressive Models (AR)	AR(p) = $y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t$, where e_t is GWN. $\phi_1, \phi_2, \dots, \phi_p$ are constants, w $\phi_p \neq 0$ When $E(y_t) = 0$, $y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t$ When $E(y_t) = \mu \neq 0$, then $y_t - \mu = \phi_1(y_{t-1} - \mu) + \phi_2(y_{t-2} - \mu) + \cdots + \phi_p(y_{t-p} - \mu) + e_t$ So $y_t = (\mu - \phi_1\mu - \phi_2\mu - \cdots - \phi_p\mu) + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t$ This is a multiple linear regression w lagged values of y_t as predictors Diff values of ϕ_k parameters results in diff TS patterns. Var of error term e_t will only change scale of series, not the patterns The defn don't guarantee process is stationary. Need to impose conditions on ϕ_k so that process is staionary
	Consider AR(1) model, $y_t = c + \phi_1 y_{t-1} + e_t$

	If $\phi_1 = c = 0$, y_t = WN. If $\phi_1 = 1$ and c = 0, y_t = random walk. If $\phi_1 = 1$ and c ≠ 0, y_t = random walk w drift. If $\phi_1 < 0$, y_t tend to oscillate btw +ve and -ve values
	Autoregressive Operator. $y_t - \phi_1 y_{t-1} - \phi_2 y_{t-2} - \cdots - \phi_p y_{t-p} = e_t$. $y_t - \phi_1 B y_t - \phi_2 B^2 y_t - \cdots - \phi_p B^p y_t = e_t$ $(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p) y_t = e_t$. So $\tau(B) y_t = e_t$, where $\tau(B)$ is a polynomial in B = autoregressive operator AR(p) is stationary if the (complex) roots of the polynomial $\tau(z)$ falls outside the unit circle
	Thrm: A linear process y_t is defined to be a linear WN e_t and is given by $y_t = \mu + \sum_{j=-\infty}^{\infty} \psi_j e_{t-j}$, where $\sum_{j=-\infty}^{\infty} \lvert\psi_j\rvert < \infty$ For a linear process, the autocovariance fn $\gamma_h = \sigma^2 \sum_{j=-\infty}^{\infty} \psi_{j+h} \psi_j$
	For AR(1), $y_t = \phi y_{t-1} + e_t$, $\tau(B) = 1 - \phi B$. So $(1 - \phi B) y_t = e_t$, $y_t = \frac{1}{1-\phi B} e_t$ From thrm, $y_t = \sum_{j=-\infty}^{\infty} \psi_j e_{t-j} = \psi_0 e_t + \psi_1 e_{t-1} + \psi_2 e_{t-2} + \cdots = \psi_0 e_t + \psi_1 B e_t + \psi_2 B^2 e_t + \cdots = (\psi_0 + \psi_1 B + \psi_2 B^2 + \cdots) e_t$ $(1 - \phi B)(\psi_0 + \psi_1 B + \psi_2 B^2 + \cdots) e_t = e_t$. So comparing coeff of B, $\psi_0 = 1$, $\psi_1 - \psi_0 \phi = 0 \Rightarrow \psi_1 = \phi$, $\psi_2 - \psi_1 \phi = 0 \Rightarrow \psi_2 = \phi^2$ In general, $\psi_j = \phi^j$. So $y_t = \sum_{j=-\infty}^{\infty} \psi_j e_{t-j} = \sum_{j=-\infty}^{\infty} \phi^j e_{t-j}$ So $\gamma_h = \sigma^2 \sum_{j=-\infty}^{\infty} \psi_{j+h} \psi_j = \sigma^2 \sum_{j=-\infty}^{\infty} \phi^{j+h} \phi^j = \sigma^2 \phi^h \sum_{j=-\infty}^{\infty} \phi^j \phi^j = \sigma^2 \phi^h \sum_{j=-\infty}^{\infty} \phi^{2j} = \sigma^2 \phi^h \frac{1}{1-\phi^2}$ (using result if |x| < 1, then $\frac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots$ and replace x with ϕ^2) OR If |ϕ| < 1, then Taylor expansion of RHS yields: $y_t = (1 + \phi B + \phi B^2 + \cdots) e_t = \sum_{j=0}^{\infty} \phi^j e_{t-j}$ $\gamma_h = cov(y_{t+h}, y_t) = E\left[\left(\sum_{j=0}^{\infty} \phi^j e_{t+h-j}\right)\left(\sum_{k=0}^{\infty} \phi^j e_{t-k}\right)\right] = E[(\phi^h \cdot 1) e_t^2 + (\phi^{h+1} \cdot \phi) e_{t-1}^2 + \cdots] = \sum_{j=0}^{\infty} E(\phi^{h+j} \phi^j e_{t-j}^2) =$ $\sigma^2 \phi^h \sum_{j=0}^{\infty} \phi^{2j} = \sigma^2 \phi^h \frac{1}{1-\phi^2}$ (using result Cov(X,Y) = E(XY) - E(X)E(Y), but both mean = 0 for this case) So $\gamma_0 = \sigma^2/(1 - \phi^2)$ and $\rho_h = \gamma_h/\gamma_0 = \phi^h$ Also $\rho_h = \begin{cases} 1 & h = 0 \\ \phi^h & h > 0 \end{cases}$
	Also, $y_t = \phi y_{t-1} + e_t \Rightarrow y_t \cdot y_{t-h} = \phi y_{t-1} \cdot y_{t-h} + e_t \cdot y_{t-h} \Rightarrow E(y_t \cdot y_{t-h}) = \phi E(y_{t-1} \cdot y_{t-h}) + E(e_t \cdot y_{t-h}) \Rightarrow \gamma_h = \phi \gamma_{h-1} + 0$ So $\gamma_h - \phi \gamma_{h-1} = 0$. $(1 - \phi B) \gamma_h = 0$
	Stationarity Conditions: For AR(1) model, require $-1 < \phi_1 < 1$. Need $\tau(z) = 1 - \phi_1 z$ to be outside unit circle. So $1 - \phi_1 z = 0$ to get $z = 1/\phi_1$. For |z| > 1, $-1 < \phi_1 < 1$ For AR(2) model, $-1 < \phi_2 < 1$ AND $\phi_1 + \phi_2 < 1$ AND $\phi_2 - \phi_1 < 1$. Use quadratic formula to find |roots| > 1
	Causal AR process. Still possible to have a stationary AR(1) process where $\phi_1 > 1$ $y_t = \phi y_{t-1} + e_t \Rightarrow y_{t+1} = \phi y_t + e_{t+1} \Rightarrow y_t = \phi^{-1} y_{t+1} - \phi^{-1} e_{t+1} = \phi^{-1}(\phi^{-1} y_{t+2} - \phi^{-1} e_{t+2}) - \phi^{-1} e_{t+1} = \cdots = \phi^{-k} y_{t+k} -$ $\sum_{j=1}^{k-1} \phi^{-j} e_{t+j}$. So comparing to linear form, $\psi_j = \phi^{-j}$ and $\sum_{j=-\infty}^{\infty} \lvert\psi_j\rvert < \infty$ So $y_t = \sum_{j=1}^{k-1} \phi^{-j} e_{t+j}$ since ϕ^{-k} tend to 0. And y_t is a linear process, hence stationary. But not casual as it is in terms of future e_t
Moving Average Models (MA)	MA(q) = $y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q}$, where e_t is WN. Each value of y_t as weighted MA of past few forecast errors Past forecast errors are used as predictors, although past errors e_t are not observed Same as AR(p) models, changing values of parameters changes behaviour of TS. Var of e_t only changes scale of series MA model here used for forecasting future values, while MA filter is to estimate trend-cycle of past values
	Moving Average Operator. Using backshift operator: $y_t = (1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q) e_t = \kappa(B) e_t$, where $\kappa(B)$ = MA operator Consider MA(1) model w mean 0: $y_t = e_t + \theta e_{t-1}$. $E(y_t) = \mu_t = 0$. Var(y_t) = $E(y_t^2) - [E(y_t)]^2 = \theta^2 E(x_{t-1}^2) + E(x_t^2) - 0 = \sigma^2(1 + \theta^2)$. So $\gamma_h = \begin{cases} (1+\theta^2)\sigma^2 & h = 0 \\ \theta\sigma^2 & h = 1. \\ 0 & h > 1 \end{cases}$ $\rho_h = \begin{cases} 1 & h = 0 \\ \theta/(1+\theta^2) & h = 1 \\ 0 & h > 1 \end{cases}$ Just as we inverted the AR(p) model earlier, we can bring the MA operator to the other side to yield (when |θ| < 1): $e_t = (1 + \theta)^{-1} y_t = \sum_{j=0}^{\infty} (-\theta)^j y_{t-j}$ = aka infinite AR representation
	Non uniqueness of MA(q) Models: Consider $y_t = 5e_{t-1} + e_t$ and $y_t = 0.2e_{t-1} + e_t$. Both have same ACF To prevent this, use MA(q) models w an infinite AR representation = invertible models General condition for invertibility is that the complex roots of $1 + \theta_1 z + \theta_2 z^2 + \cdots + \theta_q z^q$ lie outside the unit circle on the complex plane For MA(1) model: $-1 < \theta_1 < 1$ For MA(2) model, $-1 < \theta_2 < 1$ AND $\theta_1 + \theta_2 > -1$ AND $\theta_1 - \theta_2 < 1$
	Note MA model always stationary as its form is already the same as form for linear process
ARIMA Models	ARMA models combine AR(p) and MA(q) models: $y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q}$ Suppose $E(y_t) = \mu$. Then $(y_t - \mu) = \phi_1(y_{t-1} - \mu) + \cdots + \phi_p(y_{t-p} - \mu) + e_t + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q}$. So $c = \mu(1 - \phi_1 - \cdots - \phi_p)$ Predictors include both lagged values of y_t and lagged errors. Have to impose conditions on the coeff to ensure stationarity & invertibility Using backshift operator: $\tau(B) y_t = c + \kappa(B) e_t$ Model is stationary if roots of $\tau(B)$ are outside unit circle. Model is invertible if roots of $\kappa(B)$ are outside unit circle
	ARIMA = AutoRegressive Integrated Moving Average. Combine ARMA models w differencing $y_t' = c + \phi_1 y_{t-1}' + \phi_2 y_{t-2}' + \cdots + \phi_p y_{t-p}' + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q}$, where y_t' is the differenced series In backshift notation, if $(1 - B)^d y_t$ follows an ARMA model, then y_t is an ARIMA process. $(1 - B)^d \tau(B) y_t = c + \kappa(B) e_t$ Specify ARIMA model by ARIMA(p,d,q). p = order of AR part. q = order of MA part. d = degree of first differencing involved ARIMA(0,0,0) = WN. ARIMA(0,1,0) w c = 0 = RW. ARIMA(0,1,0) w c ≠ 0 = RW w drift. ARIMA(p,0,0) = AR(p). ARIMA(0,0,q) = MA(q). Alternative form: $(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p)(1 - B)^d y_t = c + (1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q) e_t$ E.g. for ARIMA(1,1,1): $(1 - \phi_1 B)(1 - B)^d y_t = c + (1 + \theta_1 B) e_t$. $(1 - \phi_1 B)$ = AR(1) part. (1 - B) = first diff part. $(1 + \theta_1 B)$ = MA(1) part
	ACF shows autocorrelations which measures r/s btw y_t and y_{t-k} for diff values of k However, if y_t and y_{t-1} are correlated, then y_{t-1} and y_{t-2} are also correlated. So y_t and y_{t-2} could be correlated simply because both are correlated to y_{t-1}. So how to measure what new info there is in y_{t-2}, that could be used in forecasting y_t Consider AR(1) process: $y_t = \phi y_{t-1} + e_t$. From earlier: $\gamma_2 = cov(y_t, y_{t-2}) = \phi^2 \gamma_0$. In fact, covariances at any positive lag h will be positive (although smaller as h incr). ACF = $\rho(h) = \phi^h$. And $\gamma(h) = cov(y_t, y_{t-h}) = E(y_t y_{t-h}) - E(y_t)E(y_{t-h}) = E(y_t y_{t-h})$ $y_{t-1} = \phi y_{t-2} + e_{t-1}$. And $y_t = \phi y_{t-1} + e_t$. And $y_{t-2} = \frac{1}{\phi}(y_{t-1} - e_{t-1})$. And $\gamma(h)/\gamma(0) = \rho(h) = \phi^h$ for AR(1) For $y_t^{1,b}$, find β that minimises $E(y_t - \beta y_{t-1})^2 = E(y_t^2 - 2\beta y_t y_{t-1} + \beta^2 y_{t-1}^2) = \gamma(0) - 2\beta\gamma(1) + \beta^2\gamma(0)$. $y_t^{1,b} = \beta y_{t-1}$ Taking derivative and setting to 0, can get $2\gamma(1) = 2\beta\gamma(0)$. So $\beta = \gamma(1)/\gamma(0) = \rho(1) = \phi$. $y_t^{1,b} = \phi y_{t-1}$ Similarly for $y_{t-2}^{1,f}$, find β that minimises $E(y_{t-2} - \beta y_{t-1})^2$ to get $\beta = \phi$. $y_{t-2}^{1,f} = \phi y_{t-1}$ So, PACF for lag 2 = $\phi_{22} = corr(y_t - y_t^{1,b}, y_{t-2} - y_{t-2}^{1,f}) = cov(y_t - \phi y_{t-1}, y_{t-2} - \phi y_{t-1}) = cov(e_t, y_{t-2} - \phi y_{t-1}) = 0$

OR $cov(y_t - \phi y_{t-1}, y_{t-2} - \phi y_{t-1}) = cov(y_t, y_{t-2}) - cov(y_t, \phi y_{t-1}) - cov(\phi y_{t-1}, y_{t-2}) + cov(\phi y_{t-1}, \phi y_{t-1}) = \gamma(2) - \phi\gamma(1) - \phi\gamma(1) + \phi^2\gamma_0 = \phi^2\gamma_0 - 2\phi^2\gamma_0 + \phi^2\gamma_0 = 0$
We have 'partialed out' the dependence on y_{t-1}. Correlation btw y_t and y_{t-2}, after accounting for effect of y_{t-1} is 0

Let $y_t^{h-1,b}$ = regression of y_t on $y_{t-1}, y_{t-2}, \dots, y_{t-h+1}$. Let $y_{t-h}^{h-1,f}$ = regression of y_{t-h} on $y_{t-h+1}, y_{t-h+2}, \dots, y_{t-1}$
Find $\beta_1, \dots, \beta_{h-1}$ s.t. $E(y_t - \beta_1 y_{t-1} - \beta_2 y_{t-2} - \cdots - \beta_{h-1} y_{t-h+1})^2$ is minimized
PACF of y_t is denoted ϕ_{hh} for h = 1,2,.... where $\phi_{11} = corr(y_t, y_{t-1})$. And $\phi_{hh} = corr(y_t - y_t^{h-1,b}, y_{t-h} - y_{t-h}^{h-1,f})$

Partial Autocorrelations measures r/s btw y_t and y_{t-k}, when effects of other time lags {1,2,3,...,k-1} are removed
Partial autocorrelation at lag k = α_k. Partial autocorrelation fn (PACF) = plot α_k for all k
α_k is computed as the estimate of ϕ_k in the AR model, $y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_k y_{t-k} + e_t$
α_k is effect of the y_{t-k}, given that all other terms are already in the model
Note that $\alpha_k = \rho_1$. For confidence bands when plotting PACF, use same critical values of $\pm 1.96/\sqrt{T}$ (same as ACF)

MA(q) ACF. MA(q) process w finite q is always stationary, and has an ACF = $\rho_h = \begin{cases} \frac{\sum_{j=0}^{q-h}\theta_j\theta_{j+h}}{1+\theta_1^2+\theta_2^2+\cdots+\theta_q^2} & 1 \le h \le q \\ 0 & otherwise \end{cases}$, i.e. cuts off after lag q

ACF and PACF behaviour
ARIMA(p,d,0) if differenced data show ACF exponentially decaying or sinusoidal AND PACF has sig spike at lag p but not beyond lag p
ARIMA(0,d,q) if differenced data show PACF exponentially decaying or sinusoidal AND ACF has sig spike at lag q but not beyond lag q

	AR(p)	MA(q)	ARMA(p,q)
ACF	Tails off	Cuts off at lag q	Tails off
PACF	Cuts off at lag p	Tails off	Tails off

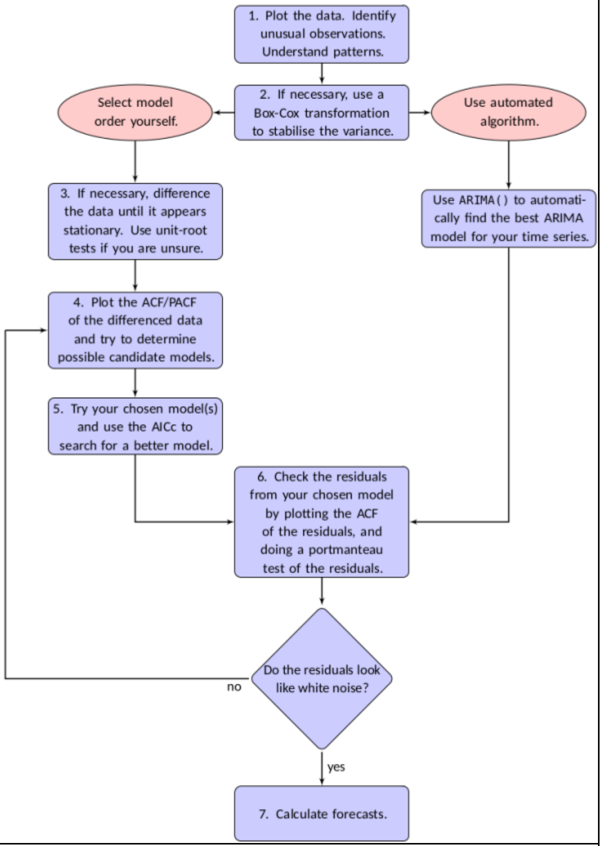
E.g. *gg_tsdisplay(TS, Y, 'partial')* # Plot time plot, ACF, PACF. If line above blue dashed line = lag is significant
If plots suggest ARIMA(3,0,0): *arima_mods <- model(TS, arima300 = ARIMA(Y ~ pdq(3,0,0) + PDQ(0,0,0)));* *report(arima_mods)*
Should also consider "nearby" candidate models. *arima_mods <- model(TS, arima300 = ..., arima201 = ...);*
fc1 <- forecast(arima_mods, h = 8);
fcast_table <- pivot_wider(fc1, 'Quarter', names_from = '.model', values_from = '.mean') %>% mutate(across(c(2,3,), ~round(.x, digits=4)))
datatable(fcast_table)

Estimation and Order Selection

For given values of p,d,q, R will find estimates for the parameters, c, ϕ_1, ... , ϕ_p, $\theta_1, \dots, \theta_q$ by maximising log-likelihood

ARIMA() in R uses a unit root tests, AIC minimisation and MLE to obtain an ARIMA model.
Default is to search through models in a step-wise manner, i.e. some models might be skipped. Can override by setting *stepwise=FALSE*
1) Determine 0 ≤ d ≤ 2 using repeated KPSS tests
2) Include constant c unless d = 2.
3) Select p and q by minimising AICc. Instead of searching through all possible models
    1. Fit 4 initial model: ARIMA(2,d,2), ARIMA(0,d,0), ARIMA(1,d,0), ARIMA(0,d,1). If d ≤ 1, fit extra model: ARIMA(0,d,0) w/o c
    2. Best model (smallest AICc value) fitted in 1. is set to be the current model
    3. Consider variations on the current model: Vary p and/or q by ±1 OR include/exclude c from the current model
4) Repeat step 3.3 until no lower AICc can be found

ARIMA Modeling Procedure.
If not using automated algo, can use the steps on the right



Point and Interval Forecasts

Point Forecasts. 1) Expand ARIMA eqn s.t. y_t is on the LHS and all other terms on RHS. 2) Rewrite eqn by replacing t w T+h
3) On RHS, replace future obs by their forecasts, future errors by 0, and past errors by the corresponding residuals
Start w h = 1, and repeat for h = 2, 3, ... until all required forecasts have been computed

E.g. Forecasts for ARIMA(3,1,1): $(1 - \hat{\phi}_1 B - \hat{\phi}_2 B^2 - \hat{\phi}_3 B^3)(1 - B) y_t = (1 + \hat{\theta}_1 B) e_t$
Then expand backshift operator on LHS to get $y_t - (1 + \hat{\phi}_1) y_{t-1} + (\hat{\phi}_1 - \hat{\phi}_2) y_{t-2} + (\hat{\phi}_2 - \hat{\phi}_3) y_{t-3} + \hat{\phi}_3 y_{t-4} = e_t + \hat{\theta}_1 e_{t-1}$
Then, $y_t = (1 + \hat{\phi}_1) y_{t-1} - (\hat{\phi}_1 - \hat{\phi}_2) y_{t-2} - (\hat{\phi}_2 - \hat{\phi}_3) y_{t-3} - \hat{\phi}_3 y_{t-4} + e_t + \hat{\theta}_1 e_{t-1}$
For h = 1, we replace t by T + 1: $y_{T+1} = (1 + \hat{\phi}_1) y_T - (\hat{\phi}_1 - \hat{\phi}_2) y_{T-1} - (\hat{\phi}_2 - \hat{\phi}_3) y_{T-2} - \hat{\phi}_3 y_{T-3} + e_{T+1} + \hat{\theta}_1 e_T$
Then replace e_{T+1} by 0 and e_T by $\hat{e}_T$: $\hat{y}_{T+1} = (1 + \hat{\phi}_1) y_T - (\hat{\phi}_1 - \hat{\phi}_2) y_{T-1} - (\hat{\phi}_2 - \hat{\phi}_3) y_{T-2} - \hat{\phi}_3 y_{T-3} + \hat{\theta}_1 \hat{e}_T$
For h = 2, forecast would be: $\hat{y}_{T+2|T} = (1 + \hat{\phi}_1) y_{T+1|T} - (\hat{\phi}_1 - \hat{\phi}_2) y_T - (\hat{\phi}_2 - \hat{\phi}_3) y_{T-1} - \hat{\phi}_3 y_{T-2}$
Note for point forecast: $\hat{y}_{T+h|T} = E(y_{T+h} | y_1, y_2, \dots, y_T)$

Computing Residuals. Consider MA(1) process: $y_t = 6 + e_t + 0.23 e_{t-1}$. Suppose y_1 = 5.2, y_2 = 6.1, y_3 = 6
Set all innovations before t = 1 to be 0. Then $y_1 = 6 + e_1 + 0.23e_0$. Then $\hat{y}_1 = E(6 + e_1 + 0.23e_0) = 6 + 0 + 0.23E(0) = 6$. So $e_1 = y_1 - \hat{y}_1 = -0.9$
$y_2 = 6 + e_2 + 0.23e_1$. $\hat{y}_2 = 6 + 0 + 0.23(-0.9) = 5.793$. $e_2 = y_2 - \hat{y}_2 = 0.307$

Forecast Intervals. Consider MA(q) model: $y_{T+h} = \mu + e_{T+h} + \theta_1 e_{T+h-1} + \theta_2 e_{T+h-2} + \cdots + \theta_q e_{T+h-q}$
When we condition on $y_1, \dots, y_T$, all innovations up to and including time T are know. They don't contribute to the variance of $y_{T+h|T}$
Hence, $var(y_{T+h|T}) = \sigma^2\left(1 + \sum_{i=1}^{h-1}\theta_i^2\right)$, where we take $\theta_i = 0$ for i > q
A 95% forecast interval for ARIMA forecasts is: $\hat{y}_{T+h|T} \pm 1.96\sqrt{v_{T+h|T}}$, where $v_{T+h|T}$ = estimated forecast var

Seasonal ARIMA models

A seasonal ARIMA model is formed by including additional seasonal terms in ARIMA models: ARIMA(p,d,q)(P,D,Q)$_m$

- (p,d,q) = non-seasonal part of model. (P,D,Q)$_m$ = seasonal part of model. m = num of obs or periods in a season

SARIMA: $(1-B^{12})^D\tau(B)y_t = c+\kappa(B)e_t \equiv$

$(1-\phi_1B-\cdots-\phi_pB^p)(1-\Phi_1B^{12}-\cdots-\Phi_PB^{12P})(1-B^{12})^Dy_t = c+(1+\theta_1B+\cdots+\theta_qB^q)(1+\Theta_1B+\cdots+\Theta_QB^{12Q})e_t$

The seasonal part of an AR or MA model can be seen in the seasonal lags of the PACF and ACF.
* Values at nonseasonal lag are 0

	SAR(P)$_m$	SMA(Q)$_m$	SARMA(P, Q)$_m$
ACF*	Tails off	Cuts off lag Qm	Tails off
PACF*	Cuts off lag Pm	Tails off	Tails off

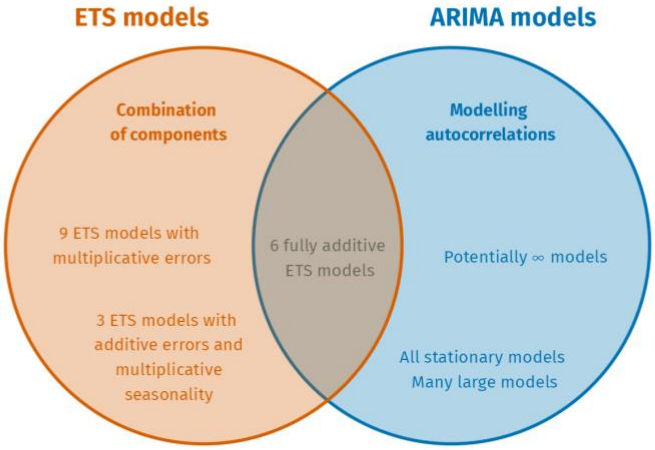
E.g. $y_t - y_{t-1} - y_{t-12} + y_{t-13} = e_t + \theta e_{t-1} + \Theta e_{t-12} + \theta\Theta e_{t-13}$
$(1-B)y_t - (1-B)B^{12}y_t = (1+\theta B)e_t + \Theta B^{12}e_t + \theta\Theta B^{12}Be_t$
$(1-B)(1-B^{12})y_t = (1+\theta B)(1+\Theta B^{12})e_t$: SARIMA(0,1,1)(0,1,1)$_{12}$

E.g. SARIMA(1,1,1)(1,1,1)$_{12}$:
$(1-\phi_1B)(1-\Phi_1B^{12})(1-B)(1-B^{12})y_t = (1+\theta B)(1+\Theta B^{12})e_t$

ETS vs ARIMA

ETS	ARIMA	Parameters
ETS(A,N,N)	ARIMA(0,1,1)	$\theta_1 = \alpha - 1$
ETS(A,A,N)	ARIMA(0,2,2)	$\theta_1 = \alpha+\beta-2.\ \ \theta_2 = 1-\alpha$
ETS(A,A$_d$,N)	ARIMA(1,1,2)	$\phi_1=\phi.\ \ \theta_1 = \alpha+\phi\beta-1-\phi.\ \ \theta_2=(1-\alpha)\phi$
ETS(A,N,A)	ARIMA(0,1,m)(0,1,0)$_m$	
ETS(A,A,A)	ARIMA(0,1,m+1)(0,1,0)$_m$	
ETS(A,A$_d$,A)	ARIMA(1,0,m+1)(0,1,0)$_m$	

ETS(A,N,N) and ARIMA(0,1,1)
$y_t = l_{t-1}+e_t.\ \ l_t = l_{t-1}+\alpha e_t \Rightarrow (1-B)l_t = \alpha e_t$
$(1-B)y_t = (1-B)l_{t-1} + (1-B)e_t \Rightarrow$
$(1-B)y_t = \alpha e_{t-1} + e_t - e_{t-1} = e_t + (\alpha-1)e_{t-1}$ = ARIMA(0,1,1)
For invertibility, $|\alpha-1|<1 \Rightarrow 0<\alpha<2$
ETS(A,A,N) and ARIMA(0,2,2)
$y_t = l_{t-1}+b_{t-1}+e_t.$
$l_t = l_{t-1}+b_{t-1}+\alpha e_t \Rightarrow (1-B)l_t = b_{t-1}+\alpha e_t$
$b_t = b_{t-1}+\beta e_t \Rightarrow (1-B)b_t = \beta e_t$
So $(1-B)^2y_t = (1-B)[(1-B)l_{t-1} + (1-B)b_{t-1} + (1-B)e_t]$
$= (1-B)[b_{t-2}+\alpha e_{t-1}+\beta e_{t-1}+(1-B)e_t]$
$= \beta e_{t-2} + (1-B)[(\alpha+\beta)e_{t-1}+e_t-e_{t-1}]$
$= \beta e_{t-2} + (\alpha+\beta-1)e_{t-1} + e_t - (\alpha+\beta-1)e_{t-2} - e_{t-1}$
$= e_t + (\alpha+\beta-2)e_{t-1} + (1-\alpha)e_{t-2}$



Tut 10

Let e_t be a WN process w var σ^2 and let $|\phi|<1$ be a constant. Consider the process $y_1 = e_1$ and $y_t = \phi y_{t-1} + e_t$, t = 2, 3, …

Note $y_t = \sum_{i=0}^{t-1}\phi^i e_{t-i}$, t = 1, 2, … . So E($y_t$) = 0 and var($y_t$) = $\sigma^2\sum_{i=1}^{t}\phi^{2(i-1)} = \sigma^2[1+\phi^2+\phi^4+\cdots+\phi^{2(t-1)}] = \sigma^2\left[\frac{1(1-\phi^{2t})}{1-\phi^2}\right]$

Since var is not constant (is a function of t), this is not a stationary process

Show for h ≥ 0: $corr(y_t,y_{t-h}) = \phi^h\left[\frac{var(y_{t-h})}{var(y_t)}\right]^{1/2}$. Note $y_t = \sum_{i=0}^{t-1}\phi^ie_{t-i}$ and $y_{t-h} = \sum_{i=0}^{t-h-1}\phi^ie_{t-h-i}$

Observe $y_t = \sum_{i=0}^{t-1}\phi^ie_{t-i} = \sum_{i=0}^{h-1}\phi^ie_{t-i} + \sum_{i=h}^{t-1}\phi^ie_{t-i} = \sum_{i=0}^{h-1}\phi^ie_{t-i} + \sum_{i=0}^{t-h-1}\phi^{i+h}e_{t-h-i} = \sum_{i=0}^{h-1}\phi^ie_{t-i} + \phi^hy_{t-h}$

So $cov(y_t,y_{t-h}) = cov(\sum_{i=0}^{h-1}\phi^ie_{t-i} + \phi^hy_{t-h},\ y_{t-h}) = cov(\phi^hy_{t-h},\ y_{t-h}) = \phi^hvar(y_{t-h})$

So $corr(y_t,y_{t-h}) = \frac{cov(y_t,y_{t-h})}{\sqrt{var(y_t)var(y_{t-h})}} = \frac{\phi^hvar(y_{t-h})}{\sqrt{var(y_t)var(y_{t-h})}} = \phi^h\left[\frac{var(y_{t-h})}{var(y_t)}\right]^{1/2}$

Argue that for large t, $var(y_t) \approx \frac{\sigma^2}{1-\phi^2}$ and $corr(y_t,y_{t-h}) \approx \phi^h$, h ≥ 0. So in a sense, y_t is asymptotically stationary

This follows since $\phi^{2t}\to 0$ as $t\to\infty$. Hence in ARIMA derivations, we work with an infinite history. By running it for a long time, process becomes stationary

Let y_t be a stationary TS w mean 0 and ACVF γ. Let a and b be constants. Suppose $x_t = a + bt + s_t + y_t$, where $s_t = s_{t-12}$ for all t
Show $(1-B)(1-B^{12})x_t$ is stationary by finding its autocovariance function in terms of γ
Let $z_t = (1-B)(1-B^{12})x_t$. Then $z_t = (1-B)(x_t - x_{t-12}) = (1-B)(a + bt + s_t + y_t - a - b(t-12) - s_{t-12} - y_{t-12}) = (1-B)(12b + y_t + y_{t-12}) = y_t - y_{t-1} - y_{t-12} + y_{t-13}$
Then E(z_t) = 0 and Cov(z_{t+h}, z_t) = Cov($y_{t+h} - y_{t+h-1} = y_{t+h-12} + y_{t+h-13}$, $y_t - y_{t-1} - y_{t-12} + y_{t-13}$) = $4\gamma(h) - 2\gamma(h+1) + \gamma(h+11) - 2\gamma(h+12) + \gamma(h+13) - 2\gamma(h-1) + \gamma(h-11) - 2\gamma(h-12) + \gamma(h-13)$
Since this is a function only of h (and not t) and because mean is constant, z_t is stationary

1. $y_t = 0.3y_{t-1} + e_t$, ARIMA(1,0,0) / AR(1) with autoregressive operator $\tau(B) = 1-0.3B$. Setting $1 - 0.3z = 0$ gives z = 10/3 > 1
Since root of $\tau(B)$ is outside unit circle, this is a stationary model. To write it as an infinite MA process:
$y_t = \frac{1}{1-0.3B}e_t = (1+0.3B+0.3^2B^2+\cdots)e_t = \sum_{j=0}^{\infty}0.3^je_{t-j}$
2. $y_t = e_t - 1.3e_{t-1} + 0.4e_{t-2}$. ARIMA(0,0,2) / MA(2) w MA operator $\kappa(B) = 1-1.3B+0.4B^2$
It is stationary since it is a linear process, w a finite set of non-zero weights.
It is also invertible since $-1 < \theta_2 < 1$; $\theta_1+\theta_2 = -0.9 > -1$ and $\theta_1 - \theta_2 = -1.7 < 1$. OR $1 - 1.3z + 0.4z^2 = 0$. so z = 1.25, 2 and outside unit circle.
3. $y_t = 0.5y_{t-1} + e_t - 1.3e_{t-1} + 0.4e_{t-2}$. ARIMA(1,0,2) w $\tau(B) = 1-0.3B$ and $\kappa(B) = 1-1.3B+0.4B^2$. SInce roots of both operators outside unit circle, it is stationary and invertible

Stationary process x_t has ACVF γ_h^x. Define new stationary series $y_t = x_t - x_{t-1}$
ACVF of y_t = $\gamma_h^y = Cov(y_{t+h},y_t) = Cov(x_{t+h}-x_{t+h-1}, x_t - x_{t-1}) = Cov(x_{t+h},x_t) - Cov(x_{t+h},x_{t-1}) - Cov(x_{t+h-1},x_t) + Cov(x_{t+h-1},x_t) = \gamma_h^x - \gamma_{h+1}^x - \gamma_{h-1}^x + \gamma_h^x = 2\gamma_h^x - \gamma_{h+1}^x - \gamma_{h-1}^x$

Consider AR(1) process w $|\phi|<1$ and mean 0: $y_t = \phi y_{t-1} + e_t$. Derive PACF for lag h = 2 using defn of PACF: $\phi_{22} = corr(y_t - y_t^{1,b}, y_{t-2} - y_{t-2}^{1,f})$
For $y_t^{1,b}$, have to find β that minimises E($y_t - \beta y_{t-1}$)^2 = E($y_t^2 - 2\beta y_ty_{t-1} + \beta^2y_{t-1}^2$). Taking derivative and setting to 0 and since this is an AR(1) process, $\beta = \gamma(1)/\gamma(0) = \rho(1) = \phi$
Similarly for $y_{t-2}^{1,f}$, minimise E($y_{t-2} - \beta y_{t-1}$)^2 to obtain $\beta = \phi$
Now PACF = $cov(y_t - \phi y_{t-1}, y_{t-2} - \phi y_{t-1}) = \gamma(2) - \phi\gamma(1) - \phi\gamma(1) + \phi^2\gamma(0) = 0$ (since for AR(1), $\gamma(h)/\gamma(0) = \phi^h$

Tut

```
features(train_set, Y, features = feature_set(tags=c("decomposition", "boxcox"))) #Suppose boxcox lambda = 0.110
ets_models <- model(train_set, ets_auto = ETS(Y));   forecast(ets_models, h=18) %>% accuracy(TS)
select(ets_models, ets_auto) %>% report();     states <- ets_models$ets_auto[[1]]$fit$states;     tail(states)
ets_after_transform <- model(train_set, ets_boxcox = ETS(box_cox(Y, 0.110)));   forecast(ets_after_transform, h=18) %>% accuracy(TS)
```

	# Decomposition model: ETS on season and RW w drift on seasonally adjusted series *model(train_set, stl1 = STL(box_cox(Y, 0.110))) %>% components %>% autoplot()* *stl_models <- model(train_set, stl1 = decomposition_model(STL(box_cox(Y, 0.110)), ETS(season_year), RW(season_adjust ~ drift())))* *forecast(stl_models, h=18) %>% accuracy(TS)*
	Suppose y_t is a mean 0, weakly stationary process with $y_t = \Phi y_{t-12} + e_t + \theta e_{t-1}, \lvert\Phi\rvert < 1, \lvert\theta\rvert < 1$ Then, $var(y_t) = var(\Phi y_{t-12} + e_t + \theta e_{t-1}) = \Phi^2 var(y_t) + \sigma^2 + \theta^2\sigma^2$. So autocovariance at lag 0 = $\gamma(0) = var(y_t) = \frac{1+\theta^2}{1-\Phi^2}\sigma^2$ Also, $(1 - B^{12}\Phi)y_t = (1 + \theta B)e_t$. SARIMA(0,0,1)(1,0,0)$_{12}$. And $y_t = (1 - B^{12}\Phi)^{-1}(1 + \theta B)e_t$ Taylor's expansion: $(1 - B^{12}\Phi)^{-1}(1 + \theta B) = (1 + \theta B)(1 + B^{12}\Phi + B^{24}\Phi^2 + \cdots) = 1 + \theta B + B^{12}\Phi + \theta\Phi B^{13} + B^{24}\Phi^2 + B^{25}\Phi^2\theta + \cdots$ So $y_t = \sum_{j=0}^{\infty} \psi_j e_{t-j}$, with $\psi_j = \begin{cases} 1 & j = 0 \\ \Phi^{j/12} & j = 12k,\ k = 1,2,3,\ldots \\ \theta\Phi^{(j-1)/12} & j = 12k+1,\ k = 1,2,3 \\ 0 & otherwise \end{cases}$ Since y_t is a linear process, $\gamma(h) = \sigma^2 \sum_{j=0}^{\infty} \psi_j\psi_{j+h}$. For lag 1, $\gamma(1) = \sigma^2 \sum_{j=0}^{\infty} \psi_j\psi_{j+1} = \sigma^2(1(\theta) + \Phi^2(\theta) + \Phi^4(\theta) + \cdots) = \sigma^2\frac{\theta}{1-\Phi^2}$. So autocorrelation with lag 1 = $\rho(1) = \frac{\gamma(1)}{\gamma(0)} = \left(\sigma^2\frac{\theta}{1-\Phi^2}\right) / \left(\frac{1+\theta^2}{1-\Phi^2}\sigma^2\right) = \frac{\theta}{1+\theta^2}$. Can also find $\rho(12) = \Phi^h$, h = 1, 2, … And $\rho(12h+1) = \frac{\theta}{1+\theta^2}\Phi^h$, h = 0, 1, 2, …

5. (6 points) Yearly water levels of a lake have been recorded from 1922 until 1972. The time plot, ACF and PACF plots for this data are given below. (*Please turn to the next page for the question.*)

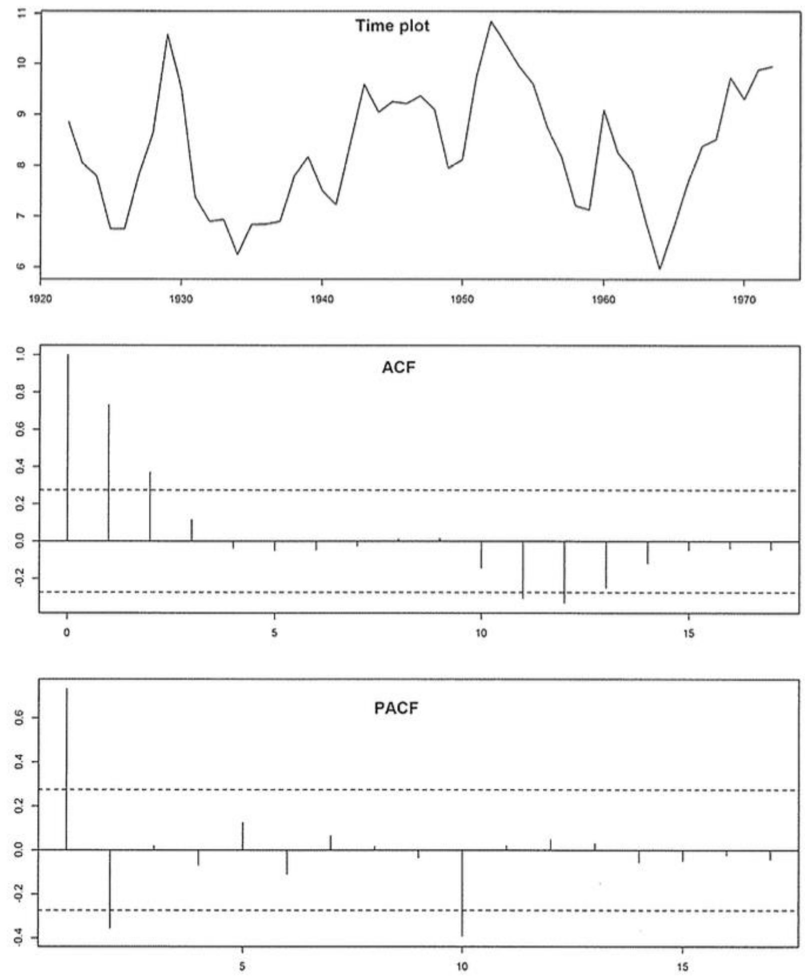


Figure 6: Lake water levels: time plot, ACF and PACF.

Suggest one ARIMA model to fit to this data
- ACF is decaying sinusoidally
- PACF cutss off after lag 10 (or 1)
- Postulate AR(1) or AR(10)