

ACTL30007 Actuarial Modelling III

Comprehensive Exam Review Notes

Omar Amin

Semester 1, 2025

Contents

1 Module 1 – Introduction to General Insurance Modelling	6
1.1 Overview of General Insurance (GI)	6
1.2 Key Definitions and Notation	6
1.2.1 Moments of Aggregate Loss	6
1.2.2 MGF and PGF Relationship	7
2 Module 2 – Collective Risk Modelling Theory	8
2.1 The Individual Risk Model	8
2.1.1 Convolution Example and basic R	8
2.2 The Collective Risk Model	9
2.3 Distribution of S	10
2.4 Distribution of S if Y is continuous	10
2.4.1 Example 12.2.3 of Bowers et al. (1997)	10
2.5 Distribution of S if Y is mixed	10
2.6 Distribution of S if Y is discrete	11
2.6.1 Example 12.2.2 of Bowers et al. (1997)	11
2.7 Distribution of S in R	11
2.7.1 Example 12.2.2 (in R)	12
2.8 Basic models for claims frequency	13
2.8.1 Compound Binomial model	13
2.8.2 Compound Poisson model	13
3 Module 3 – Individual Claim Size Modelling	15
3.1 Introduction	15
3.1.1 Steps in fitting loss models to data	15
3.1.2 Left-truncation and right-censoring	15
3.1.3 Zero claims	15
3.2 Data Analysis and Descriptive Statistics	15
3.2.1 Preliminary Steps	16
3.2.2 Visualise the SUVA raw data	16
3.2.3 Moments	17
3.2.4 SUVA Moments	18
3.2.5 Skewness and Kurtosis Plots (Cullen and Frey)	18
3.2.6 Loss Index Function	19
3.2.7 Mean-Excess Plot	20
3.2.8 Quantiles	20
3.2.9 Boxplots	21
3.2.10 Log-Log Plots	21
3.3 Parametric Models (Distributions) for Severity	22
3.3.1 Gamma distribution	22
3.3.2 Inverse Gaussian distribution	23
3.3.3 Weibull distribution	23
3.3.4 Lognormal distribution	23
3.3.5 Log-gamma distribution	23
3.3.6 Pareto distribution	24
3.4 Modelling Zero Claims	24
3.4.1 Zero-inflated severity model $X = IB$	24
3.5 Fitting of Distributions	25
3.5.1 Moment Matching Estimation (Theory and R Example)	25
3.5.2 Maximum Likelihood Estimation (Theory and R Example)	26

3.6	Parameter Uncertainty	27
3.6.1	Hessian matrix	27
3.6.2	The Wald approximation	27
3.6.3	Standard errors for SUVA data	27
3.7	Dealing with left-truncation and right-censoring	28
3.7.1	Likelihood with left-truncation and right-censoring	28
3.7.2	Coding censored data in R	28
3.7.3	Fitting censored data (in R)	29
3.7.4	Fitting left truncated data (in R)	30
3.7.5	When truncation and censoring levels are the same everywhere	31
3.7.6	When truncation and censoring levels vary	32
3.8	Model Selection	34
3.8.1	Graphical Tool Diagnostics	34
3.8.2	SUVA data Goodness of Fit Plots	35
3.8.3	Hypothesis Tests	36
3.8.4	Hypothesis Tests for Goodness-of-Fit	36
3.8.5	Using the Test Statistics	37
3.8.6	Model Selection via Information Criteria	37
3.8.7	SUVA data Hypothesis Test Statistics	37
3.9	Calculating Within Layers of Claim Size	38
3.9.1	Deductibles and Policy Limits	38
3.9.2	Useful Formulas for Truncated Moments	39
3.9.3	Reinsurance	39
3.9.4	Types of Reinsurance	39
3.9.5	Proportional Reinsurance	39
3.9.6	Non-Proportional Reinsurance	40
3.9.7	Mixed Truncation/Censoring on Retention	40
4	Module 4 – Approximations for Compound Distributions	42
4.1	The $(a, b, 0)$ class of distributions	42
4.2	The $(a, b, 1)$ class of distributions	42
4.2.1	Zero-truncated distributions	43
4.2.2	Zero-modified distributions	43
4.3	Panjer's recursion algorithm	44
4.3.1	Panjer's recursion in R	45
4.4	Approximations	45
4.4.1	Normal approximation	46
4.4.2	Translated Gamma and Log-Normal approximations	46
5	Module 5 - Copulas	48
5.1	Measures of dependence	48
5.1.1	Pearson's correlation measure	48
5.1.2	Kendall's tau	48
5.1.3	Spearman's rho	48
5.2	Copula Theory	49
5.2.1	Sklar's representation theorem	49
5.2.2	When is a Copula valid	49
5.2.3	Constructing Copulas from df's examples	49
5.2.4	Invariance property of Copulas	50
5.2.5	The Fréchet Bounds	50
5.2.6	Fundamental copulas	51
5.2.7	Survival copulas	51

5.3	Tail Dependence	51
5.3.1	Coefficient of Lower Tail Dependence	51
5.3.2	Coefficient of Upper Tail Dependence	52
5.4	Archimedean copulas	52
5.4.1	Clayton Copula	52
5.4.2	Frank copula	53
5.4.3	Gumbel Copula	53
6	Module 6 – Extreme Value Theory (EVT)	55
6.1	Measures of tail weight	55
6.2	Framework: n-block maximum	56
6.2.1	Parallel with CLT	57
6.3	GEV Distribution	57
6.3.1	Example: Exponential	57
6.3.2	Example: Pareto	58
6.3.3	GEV Density Plot of Gamma (in R)	58
6.3.4	Identifying a Fréchet GEV ($\gamma > 0$)	59
6.3.5	Identifying a Gumbel GEV ($\gamma = 0$)	59
6.3.6	Identifying a upper-bounded Weibull ($\gamma < 0$)	59
6.3.7	Estimation: Block Maxima (in R)	59
6.4	GPD Distribution	61
6.4.1	Estimation: choice of threshold u (in R)	62
7	Module 7 – Time Series Fundamentals	67
7.1	Preliminaries and White Noise Basics	67
7.1.1	Auto-covariance and correlation functions	67
7.2	Stationarity	68
7.2.1	Strictly Stationary	68
7.2.2	Weak Stationary	68
7.2.3	Properties	68
7.2.4	Trend Stationarity	69
7.2.5	Joint Stationarity	69
7.2.6	Linear processes	70
7.3	Estimation of Correlation	70
7.3.1	Sample mean	70
7.3.2	Sample autocovariance function	70
7.3.3	Sample auto-correlation function	70
7.3.4	Sample cross-covariances and cross-correlations	71
7.3.5	Prediction using cross-correlation	71
8	Module 8 - Time Series Smoothing and Exploratory Analysis	72
8.1	Regression in time series context	72
8.1.1	De-trending via regression and differencing	72
8.2	Scatter plot matrices to visualise relationships	74
8.3	Types of Smoothers (MA, Kernel, Lowess, and Splines)	75
9	Module 9 - Time Series Models	78
9.0.1	Yule-Walker Equations (and PACF)	78
9.1	AR(p) Models	79
9.1.1	Stationarity and Causality	80
9.1.2	ACF and PACF	80
9.1.3	Explosive AR models and Causality	81

9.2	MA(q) models	81
9.2.1	Non-uniqueness and Invertibility	82
9.2.2	ACF and PACF	83
9.3	ARMA(p, q) models	83
9.3.1	Parameter redundancy	83
9.3.2	Causality and (non)-future dependence	84
9.3.3	Invertibility and non-uniqueness	84
9.3.4	ACF	86
9.4	ARIMA(p, d, q) models	87
9.5	Multiplicative Seasonal ARIMA models	87
9.5.1	Pure seasonal models: ARMA(P, Q) _s	87
9.5.2	Multiplicative seasonal models: ARMA(p, q) x (P, Q) _s	89
9.6	SARIMA model ARIMA(p, d, q) x (P, D, Q) _s	90
9.7	Multivariate time series - VAR(p) models	91
9.7.1	VAR(1) model	91
9.7.2	Var(p) models	93
9.8	Auto regressive Models with Conditional Heteroscedasticity - ARCH(p) models .	96
10	Module 10 - Estimation and Forecasting	97
10.1	Behaviour of the ACF and PACF for ARMA Models	97
10.2	Behaviour of ACF and PACF in presence of seasonality	97
10.3	Box-Jenkins Methodology	97
10.3.1	Example: Air Passengers	101
10.3.2	Example: GNP data	103
10.4	Forecasting - Basics	106
10.4.1	Best Linear Predictors and Prediction for Stationary Processes	106
10.4.2	One-step-ahead prediction	107
10.5	Forecasting	108
10.5.1	Forecasts for more than one step	108
10.5.2	Predictions in practice	109
10.5.3	Truncated prediction for ARMA	109
10.5.4	Long-range forecasts	110
11	Appendix A - Midsemester Revision Sheet	111
12	Appendix B – Detailed Proofs and Derivations	114
12.1	Proof of Sklar's Theorem (Sketch)	114
12.2	Derivation of Panjer Recursion (Detailed)	114
12.3	Derivation of Special Cases of Archimedean Copulas	114
12.3.1	Clayton Copula	114
12.3.2	Frank Copula	115
12.3.3	Gumbel Copula	115
12.4	Derivation of Tail Coefficients of Archimedean Copulas	116
12.4.1	Clayton Copula	116
12.4.2	Frank Copula	116
12.4.3	Gumbel Copula	117
12.5	Derivation of Kendall's Tau of Archimedean Copulas	117
12.5.1	Clayton Copula	117
12.5.2	Gumbel Copula	118
12.6	Frank Copula	119
12.7	Derivation of GEV as Limit of Maxima	120
12.8	Derivation of ACF for AR(1) and MA(1)	120

12.8.1 AR(1)	120
12.8.2 MA(1)	121
12.9 Predictions and Graphs of log data SARIMA	121
12.10 True and False Statements for Time Series	121
13 Appendix C - General R	126
13.1 R Piping	126
13.2 Using polyroot	126
13.3 Installing tinytex	126
14 Appendix D - Practice Final Exam Worked Solutions	127
15 Appendix E - Practice MST Questions and Solutions	161
16 Appendix F - MST Questions and Solutions	167
17 Appendix G - Assignment R Code and Output	175

1 Module 1 – Introduction to General Insurance Modelling

1.1 Overview of General Insurance (GI)

General Insurance (GI), also known as non-life or property & casualty insurance, involves covering risks associated with losses (e.g., motor vehicle accidents, property damage, liability).

Unlike life insurance (where the primary event is death or survival, typically modelled by life tables and deterministic mortality laws), GI requires modelling both:

- **Frequency** of claim occurrences (how many claims happen in a fixed period, e.g., one year).
- **Severity** of claim sizes (the monetary amount of each claim).

The general framework is the *Collective Risk Model*, which defines the total loss S over a period as

$$S = \sum_{i=1}^N X_i,$$

where:

- N is the random number of claims occurring in that period (the *claim frequency*).
- X_1, X_2, \dots, X_N are the individual claim sizes (the *claim severities*), assumed i.i.d. with common distribution $F_X(x)$.
- We usually assume N is independent of the $\{X_i\}$.

Key tasks in GI modelling:

1. Estimate the distribution of N from historical claim counts (e.g., Poisson, Negative Binomial).
2. Estimate the distribution of claim sizes X (e.g., Exponential, Gamma, Lognormal, Pareto).
3. Compute or approximate the distribution of the aggregate loss S .
4. Use S to set premiums, calculate reserves, and assess risk metrics (e.g., Value-at-Risk, Tail Value-at-Risk).

In this course, we also cover:

- Dependence modelling via **copulas** (Module 5).
- **Extreme Value Theory (EVT)** for tail modelling (Module 6).
- **Time series methods** for modelling insurance data over time (Modules 7–10).

1.2 Key Definitions and Notation

Throughout these notes, we will use:

- N : claim count random variable (frequency).
- X, X_i : claim size (severity) random variable(s).
- S : aggregate loss $S = \sum_{i=1}^N X_i$.
- $F_N(n), p_N(n) = \Pr(N = n)$: mass function or distribution of N .
- $F_X(x), f_X(x)$: CDF and density (or mass function) of X .
- $\mu_N = \mathbb{E}[N], \sigma_N^2 = \text{Var}(N)$.
- $\mu_X = \mathbb{E}[X], \sigma_X^2 = \text{Var}(X)$.
- μ_S, σ_S^2 : mean and variance of S .
- $M_X(t) = \mathbb{E}[e^{tX}]$: moment generating function (MGF) of X .
- $G_N(z) = \mathbb{E}[z^N]$: probability generating function (PGF) of N .

1.2.1 Moments of Aggregate Loss

Under the assumption that N is independent of $\{X_i\}$, we have:

$$\mathbb{E}[S] = \mathbb{E}[\mathbb{E}(S | N)] = \mathbb{E}[N \cdot \mathbb{E}[X]] = \mu_N \mu_X.$$

$$\text{Var}(S) = \mathbb{E}[\text{Var}(S | N)] + \text{Var}(\mathbb{E}[S | N]).$$

Since $\text{Var}(\sum_{i=1}^N X_i \mid N) = N \sigma_X^2$ and $\mathbb{E}[S \mid N] = N \mu_X$, we get:

$$\text{Var}(S) = \mathbb{E}[N \sigma_X^2] + \text{Var}(N \mu_X) = \sigma_X^2 \mu_N + \mu_X^2 \sigma_N^2.$$

Hence,

$$\boxed{\mathbb{E}[S] = \mu_N \mu_X, \quad \text{Var}(S) = \mu_N \sigma_X^2 + \mu_X^2 \sigma_N^2.}$$

1.2.2 MGF and PGF Relationship

If $G_N(z) = \mathbb{E}[z^N]$ is the PGF of N , and $M_X(t) = \mathbb{E}[e^{tX}]$ is the MGF of X , then the MGF of S is:

$$M_S(t) = \mathbb{E}[e^{t \sum_{i=1}^N X_i}] = \mathbb{E}[(M_X(t))^N] = G_N(M_X(t)).$$

For example, if $N \sim \text{Poisson}(\lambda)$ then $G_N(z) = \exp\{\lambda(z-1)\}$. Thus

$$M_S(t) = \exp\{\lambda(M_X(t)-1)\}.$$

By expanding $M_X(t)$ as $1 + \mu_X t + \frac{1}{2}\mu_{X,2}t^2 + \dots$, we recover the moments of S above.

R Code: Simulating a Collective Risk Model

```
# Simulate aggregate loss S = sum_{i=1}^N X_i,
# where N ~ Poisson(lambda), X_i ~ Gamma(shape, scale)

lambda <- 3           # expected number of claims
shape <- 2; scale <- 1000  # Gamma parameters (mean = 2000)
nsim <- 100000
set.seed(2025)

S_sim <- numeric(nsim)
for(i in 1:nsim){
  N_i <- rpois(1, lambda)          # draw frequency
  if(N_i > 0){
    X_i <- rgamma(N_i, shape=shape, scale=scale)  # draw severities
    S_sim[i] <- sum(X_i)
  } else {
    S_sim[i] <- 0
  }
}
# Empirical mean and variance
mean(S_sim)  # ~ lambda * shape * scale = 3 * 2000 = 6000
var(S_sim)   # ~ lambda*(shape*scale^2 + shape^2*scale^2)
```

The above code simulates 10^5 years of aggregated losses. We compare the sample mean and variance to theoretical values:

$$\mathbb{E}[S] = \lambda \cdot \alpha \beta = 3 \times (2 \times 1000) = 6000,$$

$$\text{Var}(S) = \lambda(\alpha \beta^2 + \alpha^2 \beta^2) = 3(2 \times 1000^2 + (2 \times 1000)^2) = 3(2 \times 10^6 + 4 \times 10^6) = 18 \times 10^6.$$

2 Module 2 – Collective Risk Modelling Theory

Note: this module was cut short since rest is not necessary for final exam, check mid semester revision sheet for more content or refer to lecture slides.

2.1 The Individual Risk Model

In the Individual Risk Model

$$S = Y_1 + \cdots + Y_n = \sum_{i=1}^n Y_i,$$

where Y_i , $i = 1, 2, \dots, n$, are iid claims.

There are several methods to get probabilities about S :

- get the whole distribution of S (if possible)
 - Convolutions
 - Generating functions
- approximate with the help of the moments of S (Module 4)

Convolutions Formulas

In short

- discrete case:

– df:

$$F_{X+Y}(s) = \sum_x F_Y(s-x) f_X(x)$$

– pmf:

$$f_{X+Y}(s) = \sum_x f_Y(s-x) f_X(x)$$

- continuous case:

– cdf:

$$F_{X+Y}(s) = \int_{-\infty}^s F_Y(s-x) f_X(x) dx$$

– pdf:

$$f_{X+Y}(s) = \int_{-\infty}^s f_Y(s-x) f_X(x) dx$$

2.1.1 Convolution Example and basic R

Consider 3 discrete r.v.'s with probability mass functions

$$f_1(y) = \begin{cases} \frac{1}{4}, & y = 0, \\ \frac{1}{2}, & y = 1, \\ \frac{1}{4}, & y = 2, \end{cases} \quad f_2(y) = \begin{cases} \frac{1}{2}, & y = 0, \\ \frac{1}{2}, & y = 2, \end{cases} \quad f_3(y) = \begin{cases} \frac{1}{4}, & y = 0, \\ \frac{1}{2}, & y = 2, \\ \frac{1}{4}, & y = 4. \end{cases}$$

Calculate the pmf f_{1+2+3} and the df F_{1+2+3} of the sum of the three random variables.

$$\begin{aligned} f_{1+2}(2) &= \frac{1}{4} \cdot \frac{1}{2} + \frac{1}{2} \cdot 0 + \frac{1}{4} \cdot \frac{1}{2}, \\ f_{1+2+3}(4) &= \frac{1}{8} \cdot \frac{1}{4} + \frac{2}{8} \cdot 0 + \frac{2}{8} \cdot \frac{1}{2} + \frac{2}{8} \cdot 0 + \frac{1}{8} \cdot \frac{1}{4}. \end{aligned}$$

Contrary to Excel, convolutions are extremely easy to implement in R using vectors.

Table 1: Using convolutions

y	$f_1(y)$	$f_2(y)$	$f_{1+2}(y)$	$f_3(y)$	$f_{1+2+3}(y)$	$F_{1+2+3}(y)$
0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{32}$	$\frac{1}{32}$
1	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{3}{32}$	$\frac{3}{32}$
2	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{4}{32}$	$\frac{7}{32}$
3	0	0	$\frac{1}{8}$	0	$\frac{6}{32}$	$\frac{13}{32}$
4	0	0	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{3}{32}$	$\frac{19}{32}$
5	0	0	0	0	$\frac{6}{32}$	$\frac{25}{32}$
6	0	0	0	0	$\frac{4}{32}$	$\frac{29}{32}$
7	0	0	0	0	$\frac{2}{32}$	$\frac{31}{32}$
8	0	0	0	0	$\frac{1}{32}$	$\frac{32}{32}$

```

f1 <- c(1/4, 1/2, 1/4, 0, 0)
f2 <- c(1/2, 0, 1/2, 0, 0)
f12 <- c(
  f1[1] * f2[1],
  sum(f1[1:2] * f2[2:1]),
  sum(f1[1:3] * f2[3:1]),
  sum(f1[1:4] * f2[4:1]),
  sum(f1[1:5] * f2[5:1])
)
f12
## [1] 0.125 0.250 0.250 0.250 0.125

```

2.2 The Collective Risk Model

In the Collective Risk Model, aggregate losses become

$$S = Y_1 + \cdots + Y_N = \sum_{i=1}^N Y_i.$$

This is a random sum. We make the following assumptions:

- $N \in \mathbb{Z}_{\geq 0}$ is the number of claims
- $Y_i > 0$ is the amount of the i th claim
- the Y_i 's are iid with
 - (c)df $G(y)$
 - p(d/m)f $g(y)$
- the Y_i 's and N are mutually independent

We have

$$\mathbb{E}[S] = \mathbb{E}[\mathbb{E}[S | N]] = \mathbb{E}[N \mathbb{E}[Y]] = \mathbb{E}[N] \mathbb{E}[Y],$$

and

$$\begin{aligned}
 \text{Var}(S) &= \mathbb{E}[\text{Var}(S | N)] + \text{Var}(\mathbb{E}[S | N]) \\
 &= \mathbb{E}[N \text{Var}(Y)] + \text{Var}(N \mathbb{E}[Y]) \\
 &= \mathbb{E}[N] \text{Var}(Y) + \mathbb{E}[Y]^2 \text{Var}(N) \\
 &= \mathbb{E}[N](\mathbb{E}[Y^2] - \mathbb{E}[Y]^2) + \mathbb{E}[Y]^2 \text{Var}(N) \\
 &= \mathbb{E}[N] \mathbb{E}[Y^2] + \mathbb{E}[Y]^2 (\text{Var}(N) - \mathbb{E}[N]).
 \end{aligned}$$

The moment generating function of S is

$$\begin{aligned} M_S(t) &= \mathbb{E}[e^{tS}] = \mathbb{E}\left[\mathbb{E}[e^{t(Y_1+\dots+Y_N)} \mid N]\right] \\ &= \mathbb{E}[M_Y(t)^N] = \mathbb{E}[e^{N \ln M_Y(t)}] = M_N(\ln M_Y(t)). \end{aligned}$$

2.3 Distribution of S

It is possible to get a fairly general expression for the df of S by conditioning on the number of claims:

$$F_S(x) = \sum_{n=0}^{\infty} \Pr[S \leq x \mid N = n] \Pr[N = n] = \sum_{n=0}^{\infty} G^{*n}(x) \Pr[N = n] \quad (1)$$

where $G^{*n}(y)$ is the n th convolution of G .

Note that

- N will always be discrete, so this works for any type of rv Y (continuous, discrete or mixed).
- however, the type of S will depend on the type of Y .

2.4 Distribution of S if Y is continuous

If Y is continuous, S will generally be mixed:

- with a mass at 0 because of $\Pr[N = 0]$ (if positive)
- continuous elsewhere, but with a density integrating to $1 - \Pr[N = 0]$

2.4.1 Example 12.2.3 of Bowers et al. (1997)

Following from Example 12.2.1, assume now that

$$G(y) = 1 - e^{-y}, \quad M_Y(t) = \frac{1}{1-t}, \quad t < 1.$$

Now, we have that (remember $\Pr[N = 0] = p$)

$$M_S(t) = \frac{p}{1 - q M_Y(t)}.$$

It follows that

$$M_S(t) = \frac{p}{1 - q \frac{1}{1-t}} = p + q \frac{p}{p-t} = p \mathbb{E}[e^{t \cdot 0}] + (1-p) \mathbb{E}[e^{tZ}],$$

where Z is an exponential rv with parameter p . Therefore,

$$f_S(s) = \begin{cases} p, & s = 0, \\ (1-p)(p e^{-ps}), & s > 0. \end{cases}$$

2.5 Distribution of S if Y is mixed

If Y is mixed, S will generally be mixed:

- with a mass at 0 because of $\Pr[N = 0]$ and $\Pr[Y = 0]$ (if positive)
- mixed (if Y is not continuous for $x > 0$) or continuous elsewhere
- with a density integrating to something $\leq 1 - \Pr[N = 0]$

2.6 Distribution of S if Y is discrete

If Y is discrete, the pmf of S is given by:

$$f_S(s) = \sum_{n=0}^{\infty} \Pr[S = s | N = n] \Pr[N = n] = \sum_{n=0}^{\infty} g^{*n}(s) \Pr[N = n] \quad (2)$$

where $g^{*0}(0) = 1$.

- this can be implemented in a table and/or in a program.
- if the range of N goes to infinity, calculating $f_S(s)$ may require an infinite number of convolutions of Y .
- this formula is more efficient if the number of possible outcomes for N is small.
- the pmf $g^{*n}(s)$ can be calculated using de Pril's algorithm.

2.6.1 Example 12.2.2 of Bowers et al. (1997)

Using a tabular approach

(1) x	(2) $p^{*0}(x)$	(3) $p^{*1}(x) = p(x)$	(4) $p^{*2}(x)$	(5) $p^{*3}(x)$	(6) $f_S(x)$	(7) $F_S(x)$
0	1.0	—	—	—	0.1000	0.1000
1	—	0.5	—	—	0.1500	0.2500
2	—	0.4	0.25	—	0.2200	0.4700
3	—	0.1	0.40	0.125	0.2150	0.6850
4	—	—	0.26	0.300	0.1640	0.8490
5	—	—	0.08	0.315	0.0950	0.9440
6	—	—	0.01	0.184	0.0408	0.9848
7	—	—	—	0.063	0.0126	0.9974
8	—	—	—	0.012	0.0024	0.9998
9	—	—	—	0.001	0.0002	1.0000
n	0	1	2	3	—	—
$\Pr[N = n]$	0.1	0.3	0.4	0.2	—	—

- The convolutions are done the usual way.
- The number of columns depend on the range of N .
- The $f_S(x)$ are the sumproduct of row x and row $\Pr[N = n]$.

2.7 Distribution of S in R

We will make extensive use of the function `aggregateDist` from the package `actuar` (Dutang, Goulet, and Pigeon 2008):

- this function allows for several different aggregate distribution approaches, which will be introduced here (and in Module 4 as the associated theory is presented).
- here, we show how the function can be used to implement formulas (1) and (2) (using the function `convolve` in the background), which corresponds to the `method="convolution"` approach.

The basic call is

```
actuar::aggregateDist(method="convolution")
```

with the following requirements and outputs:

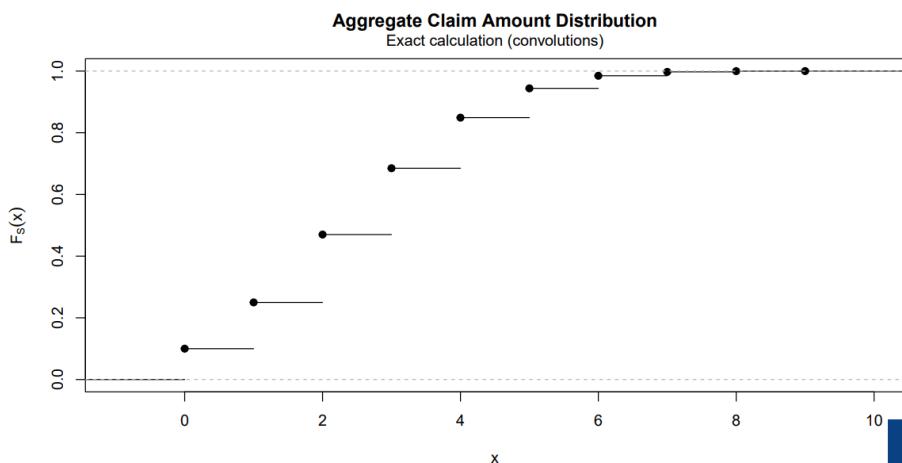
- A discrete distribution for Y is required. Note that discretisation methods are discussed in Module 4. This is input as a vector of claim-amount probability masses via the argument `model.sev=`. The first element must be $\Pr[Y = 0]$.
- There is no restriction on the shape of the frequency distribution, but it must have a finite range. This is input as a vector of claim-number probability masses via the argument `model.freq=`. The first element must be $\Pr[N = 0]$.

- The outcome of the function is the df in (1). Additional outputs:
 - `plot`: to get a pretty plot of the df
 - `summary`: to get summary statistics
 - `mean`: to get the mean
 - `diff`: to get the pmf
- Additional options are:
 - `x.scale`: currency units per unit of `sev` in the severity model (this allows calculations on multiples of \$1)

2.7.1 Example 12.2.2 (in R)

```
# Bowers 12.2.2
fy <- c(0, 0.5, 0.4, 0.1)
fn <- c(0.1, 0.3, 0.4, 0.2)
Fs <- aggregateDist("convolution", model.freq = fn, model.sev = fy)
mean(Fs)
## [1] 2.72
pmf <- c(Fs(0), diff(Fs(0:9)))
cbind(s = c(0:9), fs = pmf, Fs = Fs(0:9))
##      s      fs      Fs
## [1,] 0 0.1000 0.1000
## [2,] 1 0.1500 0.2500
## [3,] 2 0.2200 0.4700
## [4,] 3 0.2150 0.6850
## [5,] 4 0.1640 0.8490
## [6,] 5 0.0950 0.9440
## [7,] 6 0.0408 0.9848
## [8,] 7 0.0126 0.9974
## [9,] 8 0.0024 0.9998
## [10,] 9 0.0002 1.0000

summary(Fs)
## Aggregate Claim Amount Empirical CDF:
##   Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.00     2.00    3.00  2.72    4.00   9.00
plot(Fs)
```



2.8 Basic models for claims frequency

in our case, we will assume that it directly affects the likelihood of a claim to occur, i.e., the frequency, such that N/ν is normalised.

MW defines

$$p_k = \Pr[N = k], \quad k \in \mathcal{A} \subset \mathbb{N}_0,$$

where \mathcal{A} is the set of possible frequency outcomes.

there are three main assumptions for p_k :

- Binomial (with variance less than mean)
 - Poisson (with variance equal to the mean)
 - Negative binomial (a Poisson with random mean, so that variance is more than the mean)
- these all belong to a class of distributions called (a, b) , to be covered in Module 4

2.8.1 Compound Binomial model

fixed volume $\nu \in \mathbb{N}$

fixed default probability $p \in (0, 1)$ (expected claims frequency)

pmf of $N \sim \text{Binom}(\nu, p)$ is

$$p_k = \Pr[N = k] = \binom{\nu}{k} p^k (1-p)^{\nu-k}, \quad k \in \{0, \dots, \nu\} = \mathcal{A}.$$

same as a sum of Bernoulli (which is the case $\nu = 1$)

makes sense for a homogeneous portfolio with unique possible events, such as credit defaults or deaths in a life-insurance model

in R: `dbinom`, `pbinom`, `qbinom`, `rbinom`, where `size = ν` and `prob = p`

note that $\binom{\nu}{k}$ can be computed with the R function `choose`

The total claim amount S has a **compound Binomial distribution**

$$S \sim (\nu, p, G)$$

if $N \sim \text{Binom}(\nu, p)$ (with $\nu \in \mathbb{N}$, $p \in (0, 1)$) and individual claim-size distribution G .

Corollary 2.7: Assume S_1, \dots, S_n are independent with

$$S_j \sim (\nu_j, p, G), \quad j = 1, \dots, n.$$

Then

$$S = \sum_{j=1}^n S_j \sim \left(\sum_{j=1}^n \nu_j, p, G \right).$$

2.8.2 Compound Poisson model

fixed volume $\nu > 0$

expected claims frequency $\lambda > 0$

pmf of $N \sim (\lambda \nu)$ is

$$p_k = \Pr[N = k] = e^{-\lambda\nu} \frac{(\lambda\nu)^k}{k!}, \quad k \in \mathcal{A} = \mathbb{N}_0.$$

Lemma 2.9: increasing volume ν while keeping $\mathbb{E}[N] = c = \nu p$ fixed in a Binomial model leads to a Poisson distribution (more so for small p compared to ν):

$$N_\nu \xrightarrow{d} N \sim (c) \quad \text{for } N_\nu \sim (\nu, p), \quad \lim_{\nu \rightarrow \infty} \nu p = c.$$

in R: `dpois`, `ppois`, `qpois`, `rpois`, where $\lambda = \lambda \nu$

The total claim amount S has a **compound Poisson distribution**

$$S \sim (\lambda \nu, G)$$

if $N \sim (\lambda \nu)$ for given $\lambda, \nu > 0$ and individual claim-size distribution G .
the compound Poisson distribution has nice properties such as:

- the aggregation property \uparrow
- the disjoint decomposition property \downarrow

3 Module 3 – Individual Claim Size Modelling

3.1 Introduction

3.1.1 Steps in fitting loss models to data

1. **Explore and summarise data**
 - graphical explorations
 - empirical moments and quantiles
2. **Select a set of candidate distributions**
 - Pareto, log-normal, inverse Gaussian, gamma, etc.
3. **Estimate the model parameters**
 - method of moments
 - maximum likelihood (MLE)
 - maximum goodness (MGE)
4. **Evaluate the quality of a given model**
 - graphical procedures (QQ, PP plots, empirical CDFs)
 - score-based approaches (Kolmogorov–Smirnov tests, AD tests, chi-square goodness-of-fit tests)
 - likelihood-based information criteria (AIC, BIC)
5. **Determine which model to choose based on needs**

3.1.2 Left-truncation and right-censoring

- **Left-truncated observation** (e.g. excess / deductible) – exact value only recorded if it fulfils a certain condition:
 - An observation is left-truncated at c if it is *not* recorded when it is at or below c , and if it exceeds c it is recorded exactly.
- **Right-censored observation** (e.g. policy limit) – exact value not known:
 - An observation is right-censored at d if it is recorded as d when at or above d , and if below d it is recorded exactly.
- Similarly, one can define right-truncation, left-censoring, etc.
- Observations may be both left-truncated and right-censored, as often occurs in actuarial data.

3.1.3 Zero claims

- Significant proportions of **zero claims** are frequent, for a number of reasons:
 - data is policy-based, not claims-based;
 - claims not exceeding deductible;
 - mandatory reporting of accidents;
 - “bonus hunger,” etc.
- This complicates fitting, since many parametric severity distributions do not allow a flexible mass at 0.
- Several possible solutions:
 - adjust Y by mixing in a point mass at 0 with a parametric distribution;
 - adjust the claim frequency model (i.e. ignore zeros), but this
 - * may underestimate volatility (the proportion of zeros may itself be random),
 - * should be avoided if zeros represent claims of no cost (rather than absence of a claim).

3.2 Data Analysis and Descriptive Statistics

```
# Using SUVA datasets from lectures
SUVA <- read_excel("SUVA.xls")
as_tibble(SUVA)
# This data set has 2 columns "medcosts" and "dailyallow"
```

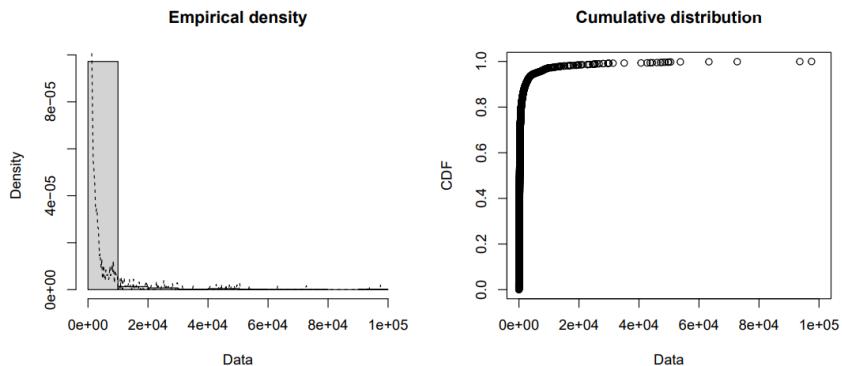
3.2.1 Preliminary Steps

Before any modelling is done, it is essential to gain a preliminary understanding of the data:

- **Summarise the data:**
 - summary statistics: mean, median, standard deviation, coefficient of variation, skewness, quantiles, min, max, etc.
- **Visualise the data:**
 - histogram with kernel density overlay,
 - empirical CDF, compared to a theoretical CDF via QQ or PP plot.
- For heavy-tailed data, repeat the above on the log-scale (e.g. compare to lognormal).
- Understand data collection procedures and standards.
- Investigate any unusual features (outliers, breaks, etc.) by consulting claim adjusters or data owners if possible.

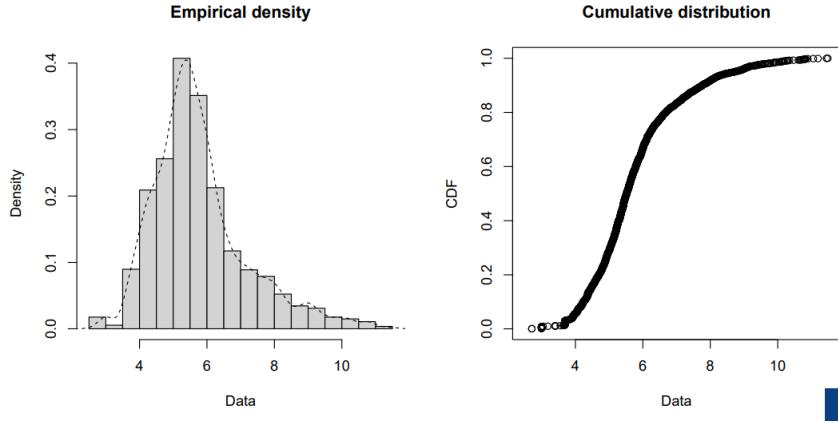
3.2.2 Visualise the SUVA raw data

```
fitdistrplus::plotdist(SUVA$medcosts, histo = TRUE, demp = TRUE)
sum(SUVA$medcosts > 0)/length(SUVA$medcosts)
## [1] 0.966896
```



Log of raw SUVA data

```
fitdistrplus::plotdist(log(SUVA$medcosts[SUVA$medcosts > 0]), histo = TRUE,
demp = TRUE)
```



- Medical costs remain skewed even on the log scale, suggesting an extremely heavy-tailed distribution may be necessary.
- Daily allowance costs look roughly symmetric after logging, suggesting a lognormal (or similar) model might be appropriate.
- Removing zeros is crucial for the daily allowance data (and for taking logarithms), since more than half of the claims are zero.

3.2.3 Moments

Moments of a distribution provide information:

- The first moment is the mean, which provides its location.
- The second *central* moment is the variance (and is related to the coefficient of variation), giving an idea of dispersion around the mean.
- The third *standardized* moment is the skewness of the data.
- The fourth *standardized* moment is the kurtosis, which indicates tail-fatness.
 - Excess kurtosis = kurtosis – 3, measured relative to the normal distribution.

Loss size index and mean excess function

The following summary functions are also helpful:

- **Loss-size index function** of y :

$$\mathcal{I}(G(y)) = \frac{\int_0^y z dG(z)}{\int_0^\infty z dG(z)}, \quad \widehat{\mathcal{I}}_n(\alpha) = \frac{\sum_{i=1}^{\lfloor n\alpha \rfloor} Y_{(i)}}{\sum_{i=1}^n Y_i}, \quad \alpha \in [0, 1].$$

This measures the contribution of $[0, y]$ to the overall mean (cf. Pareto principle: 80% of cost from top 20% of claims).

- **Mean excess function** of u :

$$e(u) = E[Y - u | Y > u], \quad \widehat{e}_n(u) = \frac{\sum_{i=1}^n (Y_i - u) \mathbf{1}_{\{Y_i > u\}}}{\sum_{i=1}^n \mathbf{1}_{\{Y_i > u\}}}.$$

An increasing $e(u)$ indicates a heavy-tailed distribution (see Module 6). Note $e(0)$ is the mean when all claims are strictly positive.

R and Technical Notes:

- To compute empirical quantities:
 - `actuar::emm` provides empirical moments of any order.

- `mean`, `var`, `sd` give the (unbiased) sample mean, variance, and standard deviation.
- Code for the loss-size index function is provided in the illustration.
- Code for the mean-excess function is also provided, though one often uses `extRemes::mrlplot()` for plotting.
- The function `fitdistrplus::descdist()` produces a skewness–kurtosis plot showing the empirical point relative to the theoretical region for various distributions.
 - The `boot` parameter enables non-parametric bootstrap of the moment estimates to assess variability.
 - The `method` argument may be "unbiased" or "sample" to switch between unbiased or sample moments.

3.2.4 SUVA Moments

Medical Costs:

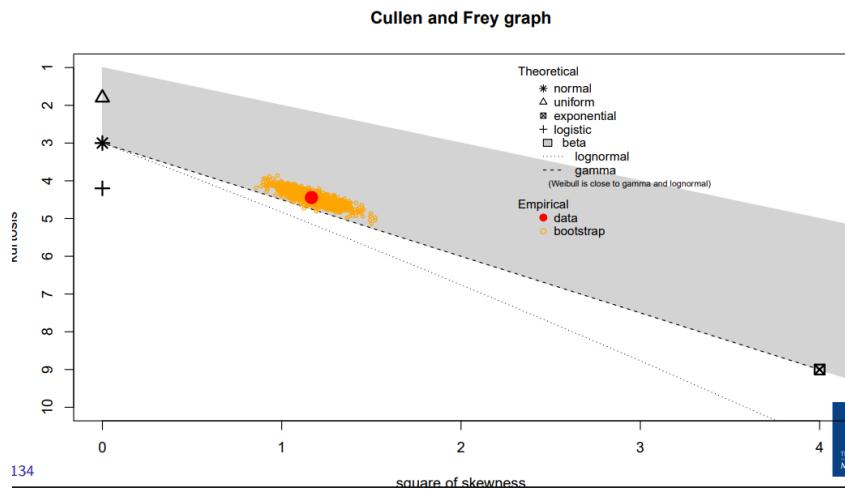
```
format(actuar::emm(SUVA$medcosts, order = 1:3), scientific = FALSE)
## [1] " 1443.349" " 34268506.007" "1791560934502.502"
sd(SUVA$medcosts)/mean(SUVA$medcosts)
## [1] 3.93143
```

Daily Allowance:

```
format(actuar::emm(SUVA$dailyallow, order = 1:3), scientific = FALSE)
## [1] " 3194.15" " 172677852.63" "20364647975482.07"
sd(SUVA$dailyallow)/mean(SUVA$dailyallow)
## [1] 3.991459
```

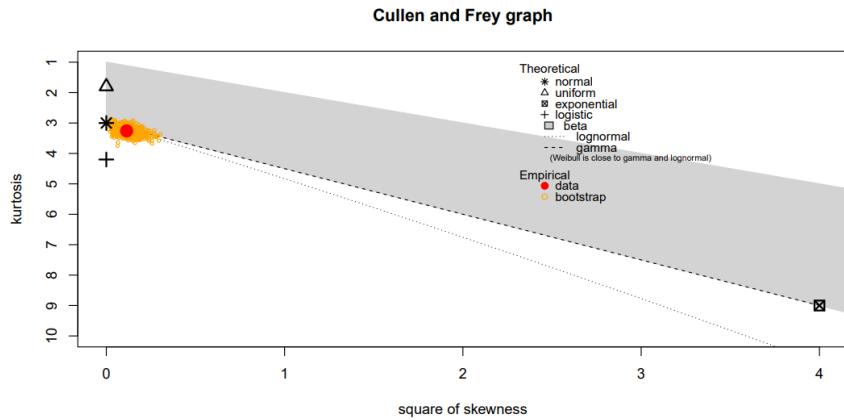
3.2.5 Skewness and Kurtosis Plots (Cullen and Frey)

```
fitdistrplus::descdist(log(SUVA$medcosts[SUVA$medcosts > 0]), boot = 1000)
```



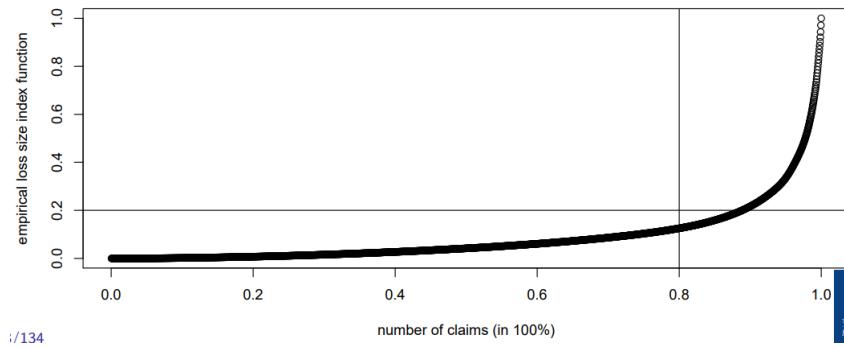
```
fitdistrplus::descdist(log(SUVA$dailyallow[SUVA$medcosts > 0]), boot = 1000)
## summary statistics
## -----
## min: 3.258097 max: 12.13806
## median: 7.474772
## mean: 7.63051
## estimated sd: 1.441014
## estimated skewness: 0.3378648
```

```
## estimated kurtosis: 3.259708
```

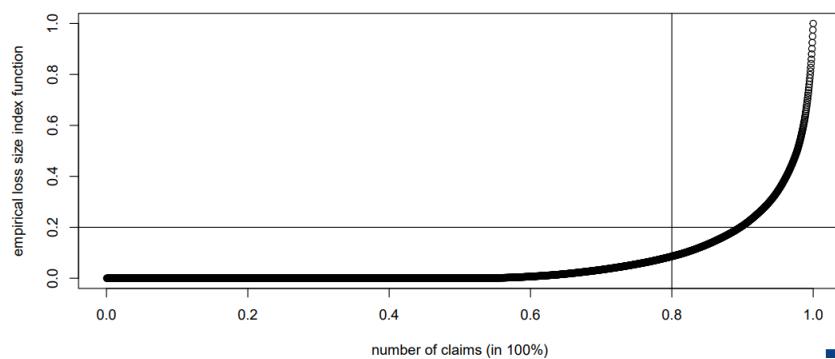


3.2.6 Loss Index Function

```
SUVA.MC.lif <- cumsum(sort(SUVA$medcosts))/sum(SUVA$medcosts)
plot(1:length(SUVA$medcosts)/length(SUVA$medcosts),
     SUVA.MC.lif, xlab = "number of claims (in 100%)",
     ylab = "empirical loss size index function")
abline(h = 0.2, v = 0.8)
```



```
SUVA.DA.lif <- cumsum(sort(SUVA$dailyallow))/sum(SUVA$dailyallow)
plot(1:length(SUVA$dailyallow)/length(SUVA$dailyallow),
     SUVA.DA.lif, xlab = "number of claims (in 100%)",
     ylab = "empirical loss size index function")
abline(h = 0.2, v = 0.8)
```



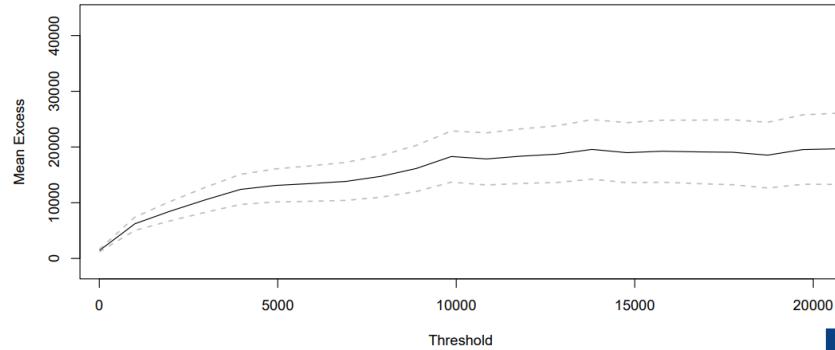
3.2.7 Mean-Excess Plot

This function will return the mean excess function for an arbitrary vector of thresholds u (for instance, 0, 100, 1000 and 10000 here)

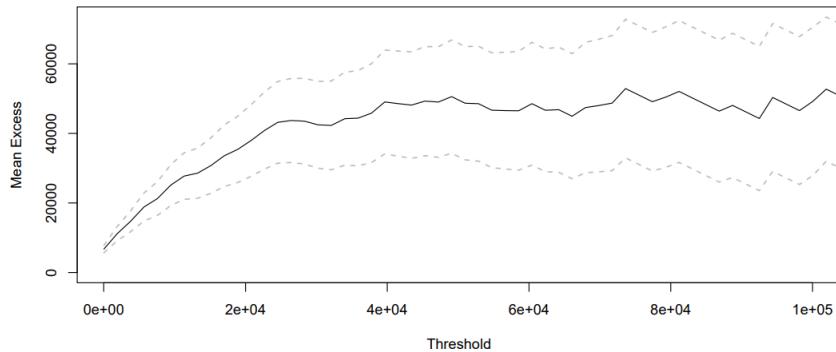
```
mef <- function(x, u) {
  mefvector <- c()
  for (i in u) {
    mefvector <- c(mefvector, sum(pmax(sort(x) - i, 0))/length(x[x > i]))
  }
  return(mefvector)
}
mef(SUVA$medcosts, c(0, 100, 1000, 10000))
## [1] 1492.765 1709.148 6233.296 18156.631
mean(SUVA$medcosts)
## [1] 1443.349
mean(SUVA$medcosts[SUVA$medcosts > 0])
## [1] 1492.765
```

This graph is done easily using `extRemes::mrlplot` (a linear increase means its heavy tailed):

```
mrlplot(SUVA$medcosts[SUVA$medcosts > 0], xlim = c(250, 20000))
# If you restrict the x range the graph isn't biased by a few very large values
```



```
mrlplot(SUVA$dailyallow[SUVA$dailyallow > 0], xlim = c(250, 1e+05))
# again restrict xlim, because it shows its not as heavy tail as the whole domain says
# its kind of flat
```



3.2.8 Quantiles

```

quantile(SUVA$dailyallow[SUVA$dailyallow > 0])
##   0%    25%    50%    75%   100%
## 26.0 842.0 1763.0 4841.5 186850.0

##### could also focus on particular quantiles
quantile(SUVA$medcosts[SUVA$medcosts > 0], probs = c(0.9, 0.95, 0.97, 0.99))
##   90%    95%    97%    99%
## 2351.00 6120.20 9661.08 27450.60
quantile(SUVA$dailyallow[SUVA$dailyallow > 0], probs = c(0.9, 0.95, 0.97, 0.99))
##   90%    95%    97%    99%
## 14015.0 25140.5 41896.7 93285.0

```

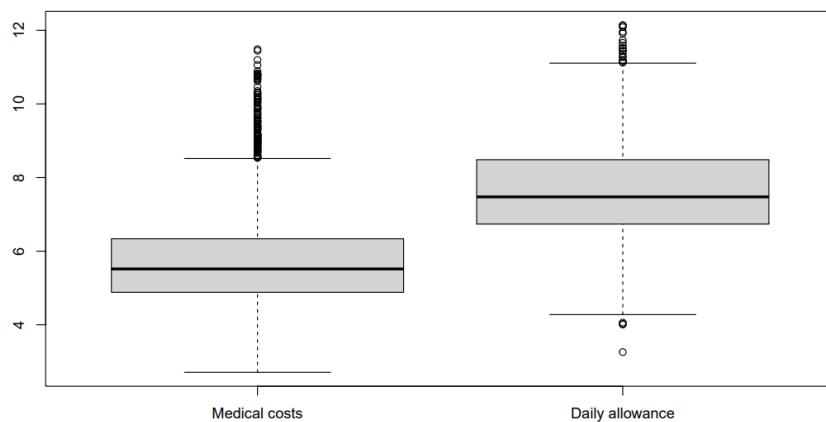
This is a crude way of estimating Value at Risk (VaR)

3.2.9 Boxplots

```

boxplot(list(`Medical costs` = log(SUVA$medcosts[SUVA$medcosts > 0]),
`Daily allowance` = log(SUVA$dailyallow[SUVA$dailyallow > 0])))
# the unlogged version is too bad to see anything

```



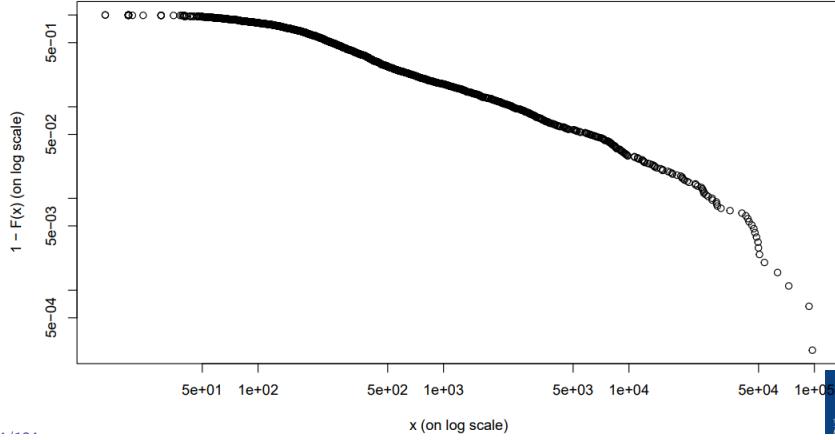
3.2.10 Log-Log Plots

The log-log plot is defined as

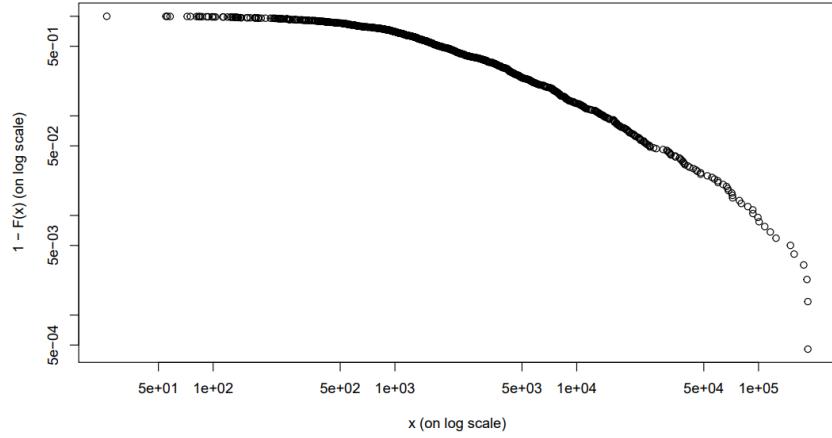
$$y \mapsto [\ln y, \ln(1 - G(y))],$$

where G is replaced by \hat{G} in the empirical version. Just as for the empirical mean-excess plot, a linear (now decreasing) pattern in the empirical log-log plot suggests a heavy-tailed distribution.

```
emplot(SUVA$medcosts[SUVA$medcosts > 0], alog = "xy", labels = TRUE)
```



```
emplot(SUVA$dailyallow[SUVA$dailyallow > 0], alog = "xy", labels = TRUE)
```



Again, medical costs are a good candidate for heavy tailed distributions (graph is more linear, except at the very end), whereas daily allowance is more reasonably behaved (graph is more concave).

3.3 Parametric Models (Distributions) for Severity

3.3.1 Gamma distribution

Let $Y \sim \Gamma(\alpha, \beta)$ with density

$$g(y) = \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} e^{-\beta y}, \quad y > 0, \alpha, \beta > 0.$$

- CDF:

$$G(y) = \frac{1}{\Gamma(\alpha)} \int_0^{\beta y} t^{\alpha-1} e^{-t} dt = P(\alpha, \beta y),$$

where P is the regularized lower incomplete gamma.

- **Mean:** $[Y] = \alpha/\beta$.
- **Variance:** $(Y) = \alpha/\beta^2$.
- **Skewness:** $\gamma_1(Y) = 2/\sqrt{\alpha}$.
- **MGF:** $M_Y(t) = \left(\frac{\beta}{\beta-t}\right)^\alpha, t < \beta$.
- **Moments:** $[Y^k] = \Gamma(\alpha + k) / (\Gamma(\alpha) \beta^k)$.
- **Special case:** $\alpha = 1 \implies Y \sim (\beta)$.

3.3.2 Inverse Gaussian distribution

Let $Y \sim (\alpha, \beta)$ with density

$$g(y) = \frac{\alpha y^{-3/2}}{\sqrt{2\pi\beta}} \exp\left[-\frac{(\alpha - \beta y)^2}{2\beta y}\right], \quad y > 0, \alpha, \beta > 0.$$

- **Mean:** $[Y] = \alpha/\beta$.
- **Variance:** $(Y) = \alpha/\beta^2$.
- **Skewness:** $\gamma_1(Y) = 3/\sqrt{\alpha}$.
- **MGF:** $M_Y(t) = \exp[\alpha(1 - \sqrt{1 - 2t/\beta})]$, $t < \beta/2$.
- **CDF:** no simple closed form; expressible via standard normal CDFs.

3.3.3 Weibull distribution

Let $Y \sim (\tau, c)$ with density

$$g(y) = (c\tau)(cy)^{\tau-1}e^{-(cy)^\tau}, \quad y > 0, c, \tau > 0.$$

- **CDF:** $G(y) = 1 - \exp\{-(cy)^\tau\}$.
- **Mean:** $[Y] = \Gamma(1 + 1/\tau)/c$.
- **Variance:** $(Y) = \Gamma(1 + 2/\tau)/c^2 - \mu_Y^2$.
- **Skewness:** $\gamma_1(Y) = [\Gamma(1 + 3/\tau)/c^3 - 3\mu_Y\sigma_Y^2 - \mu_Y^3]/\sigma_Y^3$.
- **MGF:** does not exist for $\tau < 1$ and $t > 0$.
- **Moments:** $[Y^k] = \Gamma(1 + k/\tau)/c^k$.
- **Note:** If $Z \sim (1)$, then $Z^{1/\tau}/c \sim (\tau, c)$.

3.3.4 Lognormal distribution

Let $Y \sim (\mu, \sigma^2)$, so $\ln Y \sim (\mu, \sigma^2)$.

- **CDF:** $\Pr(Y \leq y) = \Phi((\ln y - \mu)/\sigma)$.
- **Mean:** $[Y] = \exp\{\mu + \sigma^2/2\}$.
- **Variance:** $(Y) = \exp\{2\mu + \sigma^2\}(\exp\{\sigma^2\} - 1)$.
- **Skewness:** $(\exp\{\sigma^2\} + 2)\sqrt{\exp\{\sigma^2\} - 1}$.
- **MGF:** does not exist for $t > 0$.

3.3.5 Log-gamma distribution

If $\ln Y \sim \Gamma(\gamma, c)$, then Y has density

$$g(y) = \frac{c^\gamma}{\Gamma(\gamma)} (\ln y)^{\gamma-1} y^{-(c+1)}, \quad y > 0, \gamma, c > 0.$$

- **CDF:** $G(y) = \frac{1}{\Gamma(\gamma)} \int_0^{\ln y} t^{\gamma-1} e^{-ct} dt$.
- **Mean:** $[Y] = (c/(c-1))^\gamma, c > 1$.
- **Variance:** $(Y) = (c/(c-2))^\gamma - \mu_Y^2, c > 2$.
- **Skewness:** $[(c/(c-3))^\gamma - 3\mu_Y\sigma_Y^2 - \mu_Y^3]/\sigma_Y^3, c > 3$.
- **MGF:** does not exist for $t > 0$.
- **Moments:** $[Y^k] = (c/(c-k))^\gamma, c > k$.

3.3.6 Pareto distribution

Let $Y \sim (\theta, \alpha)$ with density

$$g(y) = \frac{\alpha}{\theta} \left(\frac{y}{\theta}\right)^{-(\alpha+1)}, \quad y \geq \theta, \quad \alpha > 0.$$

- **CDF:** $G(y) = 1 - (\theta/y)^\alpha$, $y \geq \theta$.
- **Mean:** $[Y] = \theta \alpha / (\alpha - 1)$, $\alpha > 1$.
- **Variance:** $(Y) = \theta^2 \frac{\alpha}{(\alpha-1)^2(\alpha-2)}$, $\alpha > 2$.
- **Skewness:** $\gamma_1(Y) = \frac{2(1+\alpha)/(\alpha-3)}{\sqrt{(\alpha-2)/\alpha}}$, $\alpha > 3$.
- **MGF:** does not exist for $t > 0$.

3.4 Modelling Zero Claims

3.4.1 Zero-inflated severity model $X = IB$

In this approach $X = IB$, where

- I is an indicator of whether a claim occurs, with

$$\Pr[I = 1] = q, \quad \Pr[I = 0] = 1 - q.$$

- B is the claim amount given $I = 1$.

Resulting distribution

$$\Pr[X \leq x] = \Pr[X \leq x | I = 0] \Pr[I = 0] + \Pr[X \leq x | I = 1] \Pr[I = 1] = (1 - q) + q \Pr[B \leq x].$$

Hence the moment generating function is

$$M_X(t) = \mathbb{E}[e^{tX}] = \mathbb{E}[e^{tX} | I = 0] (1 - q) + \mathbb{E}[e^{tX} | I = 1] q = (1 - q) + q \mathbb{E}[e^{tB}].$$

Mean and Variance

By conditioning,

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X | I]] = \mathbb{E}[I \mathbb{E}[B]] = q \mathbb{E}[B],$$

and

$$\text{Var}(X) = \text{Var}(\mathbb{E}[X | I]) + \mathbb{E}[\text{Var}(X | I)] = (\mathbb{E}[B])^2 \text{Var}(I) + \mathbb{E}[I^2] \text{Var}(B) = q(1-q)(\mathbb{E}[B])^2 + q \text{Var}(B).$$

Special case: $X = Ib$

- Fixed claim amount $B = b$ w.p. 1.
- Then $X = Ib$ takes

$$X = \begin{cases} b, & I = 1, \\ 0, & I = 0, \end{cases}$$

so $\mathbb{E}[X] = bq$ and $\text{Var}(X) = b^2 q(1 - q)$. In other words, X is a scaled Bernoulli.

Example: Bicycle Theft (retrieving B from X)

An insurer covers bicycle theft up to \$400, but pays only half if the bike was not locked. Suppose

$$\Pr[X = 400] = 0.05, \quad \Pr[X = 200] = 0.15, \quad q = \Pr[I = 1] = 0.20.$$

Then for $x = 400$ we back out

$$\Pr[B = 400] = \Pr[X = 400 | I = 1] = \frac{\Pr[X = 400 \cap I = 1]}{\Pr[I = 1]} = \frac{0.05}{0.20} = 0.25.$$

3.5 Fitting of Distributions

Some fitting criteria include:

- moment matching: choose the m parameters such that the first m moments match
- maximum likelihood: choose the parameters such that the overall likelihood that the model generated the data is maximal

R and Technical Notes:

- MASS:::fitdist is standard, and uses by default MLE via the `optim` function.
- `fitdistrplus` allows other fitting criteria (such as method of moments and maximum goodness), and also allows for the user to supply their own optimisation function.
- distributions are coded in the following way, e.g., for distribution `foo`:
 - `dfoo` is the density (pdf)
 - `pfoo` is the distribution function (cdf)
 - `qfoo` is the quantile function
 - `rfoo` is the random number generator

3.5.1 Moment Matching Estimation (Theory and R Example)

This is quite straightforward—for a distribution with m parameters:

- Choose m moments you care about (usually the first m moments about the origin or the mean).
- Build a system of m equations (one per parameter) equating the theoretical moments to the empirical moments.
- Solve the system (analytically if possible, otherwise numerically).

In R, set the argument `method="mme"` in the call to `fitdist`:

- For 1- and 2-parameter distributions, the mean and variance are matched in closed form.
- For some other distributions, closed-form solutions for higher moments are used.
- Otherwise, the equations are solved numerically (via `optim`), minimising the sum of squared differences between theoretical and empirical moments.

R SUVA Data Example

```
fit.lnorm.mme <- fitdistrplus::fitdist(log(SUVA$dailyallow[SUVA$dailyallow > 0]),
                                         "lnorm", method = "mme", order = 1:2)
fit.lnorm.mme$estimate
##   meanlog      sdlog
## 2.0146490 0.1871132

fit.lnorm.mme$loglik
## [1] -1959.06

fit.gamma.mme <- fitdist(log(SUVA$dailyallow[SUVA$dailyallow > 0]),
                           "gamma", method = "mme", order = 1:2)
fit.gamma.mme$estimate
##   shape      rate
## 28.065081 3.678009

fit.gamma.mme$loglik
## [1] -1951.838

# function to calculate sample raw moment
memp <- function(x, order) emm(x, order)
fit.weibull.mme <- fitdist(log(SUVA$dailyallow[SUVA$dailyallow > 0]),
                            "weibull", method = "mme", memp = memp, order = c(1, 2))
```

```

fit.weibull.mme$estimate
##      shape      scale
## 6.172910 8.212158

fit.weibull.mme$loglik
## [1] -2020.119

```

3.5.2 Maximum Likelihood Estimation (Theory and R Example)

The likelihood for a statistical model indicates how probable the observed data are under the model. Without censoring or truncation, for observations $\mathbf{x} = (x_1, \dots, x_n)$ and parameter vector $\theta = (\theta_1, \dots, \theta_m)$,

$$L(\theta; \mathbf{x}) = \prod_{i=1}^n f(x_i; \theta).$$

The value $\hat{\theta}$ that maximizes $L(\theta; \mathbf{x})$ is the *maximum likelihood estimator*. It is more convenient to maximize the log-likelihood

$$\ell(\theta; \mathbf{x}) = \log L(\theta; \mathbf{x}),$$

which avoids numerical underflow when n is large. MLEs enjoy properties such as consistency and asymptotic normality, allowing standard errors to be estimated via the observed information matrix.

R SUVA Data Example

```

fit.lnorm.mle <- fitdist(log(SUVA$dailyallow[SUVA$dailyallow > 0]),
  "lnorm", method = "mle")
fit.lnorm.mle$estimate
##    meanlog      sdlog
## 2.0140792 0.1918442

fit.lnorm.mle$loglik
## [1] -1958.359

fit.gamma.mle <- fitdist(log(SUVA$dailyallow[SUVA$dailyallow > 0]),
  "gamma")
fit.gamma.mle$estimate
##      shape      rate
## 27.819839 3.646072

fit.gamma.mle$loglik
## [1] -1951.818

fit.weibull.mle <- fitdist(log(SUVA$dailyallow[SUVA$dailyallow > 0]),
  "weibull")
fit.weibull.mle$estimate
##      shape      scale
## 5.527347 8.231313

fit.weibull.mle$loglik
## [1] -2003.26

summary(fit.weibull.mle)
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters :
##      estimate Std. Error

```

```

## shape 5.527347 0.12127037
## scale 8.231313 0.04760155
## Loglikelihood: -2003.26 AIC: 4010.519 BIC: 4020.524
## Correlation matrix:
##           shape      scale
## shape 1.0000000 0.3301744
## scale 0.3301744 1.0000000

```

3.6 Parameter Uncertainty

3.6.1 Hessian matrix

The *score* (or gradient) vector consists of the first derivatives of the log-likelihood $\ell(\theta; \mathbf{x})$:

$$S(\theta; \mathbf{x}) = \left(\frac{\partial \ell(\theta; \mathbf{x})}{\partial \theta_1}, \dots, \frac{\partial \ell(\theta; \mathbf{x})}{\partial \theta_m} \right)',$$

so that the MLE $\hat{\theta}$ satisfies the first-order condition

$$S(\hat{\theta}; \mathbf{x}) = \mathbf{0}.$$

The $m \times m$ *Hessian* matrix is the matrix of second derivatives:

$$H(\theta; \mathbf{x}) = \frac{\partial^2 \ell(\theta; \mathbf{x})}{\partial \theta \partial \theta'} = \begin{pmatrix} \frac{\partial^2 \ell(\theta; \mathbf{x})}{\partial \theta_1^2} & \cdots & \frac{\partial^2 \ell(\theta; \mathbf{x})}{\partial \theta_1 \partial \theta_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \ell(\theta; \mathbf{x})}{\partial \theta_m \partial \theta_1} & \cdots & \frac{\partial^2 \ell(\theta; \mathbf{x})}{\partial \theta_m^2} \end{pmatrix}.$$

The *Fisher information* is the (expected) variance of the score:

$$\mathcal{I}(\theta) = -\mathbb{E}[H(\theta; \mathbf{x})].$$

3.6.2 The Wald approximation

A consistent estimator of the covariance matrix of $\hat{\theta}$ is given by the inverse of the estimated Fisher information at $\hat{\theta}$:

$$\text{Var}(\hat{\theta}) \approx [-\mathbb{E}(H(\hat{\theta}; \mathbf{x}))]^{-1}.$$

The square roots of the diagonal entries of this matrix are the standard errors of the MLE. Here:

- “Consistent” means convergence in probability to the true value.
- As $n \rightarrow \infty$, $\hat{\theta}$ is asymptotically normal.
- These are asymptotic results, and in practice depend on the data, the model, and the parametrisation.

3.6.3 Standard errors for SUVA data

```

# You can't get this for MME, only MLE:
fit.lnorm.mle$sd
##      meanlog      sdlog
## 0.005786951 0.004091492

fit.gamma.mle$sd
##      shape      rate
## 1.1797330 0.1560157

```

```

fit.weibull.mle$sd
##      shape      scale
## 0.12127037 0.04760155

```

3.7 Dealing with left-truncation and right-censoring

3.7.1 Likelihood with left-truncation and right-censoring

Suppose we observe triplets (t_j, x_j, δ_j) , $j = 1, \dots, n$, where

- t_j is the left-truncation threshold (data are only recorded if $X > t_j$),
- x_j is the recorded value (either the exact loss or the censoring limit),
- δ_j is the right-censoring indicator:

$$\delta_j = \begin{cases} 0, & \text{if the observation is exact (limit not reached),} \\ 1, & \text{if right-censored (limit reached).} \end{cases}$$

If the underlying model has pdf $f(\cdot; \theta)$ and cdf $F(\cdot; \theta)$, then the contribution of observation j to the likelihood is

$$L_j(\theta) = \left[\frac{f(x_j; \theta)}{1 - F(t_j; \theta)} \right]^{1-\delta_j} \times \left[\frac{1 - F(x_j; \theta)}{1 - F(t_j; \theta)} \right]^{\delta_j}.$$

Hence the full likelihood is

$$L(\theta) = \prod_{j=1}^n \left[\frac{f(x_j; \theta)}{1 - F(t_j; \theta)} \right]^{1-\delta_j} \times \left[\frac{1 - F(x_j; \theta)}{1 - F(t_j; \theta)} \right]^{\delta_j}.$$

Equivalently:

- If $\delta_j = 0$ (exact), $L_j(\theta) = f(x_j; \theta)/(1 - F(t_j; \theta))$.
- If $\delta_j = 1$ (censored), $L_j(\theta) = [1 - F(x_j; \theta)]/(1 - F(t_j; \theta))$.

Note: if a policy limit u_j is reached, one often sets $x_j = t_j + u_j$.

3.7.2 Coding censored data in R

Prepare a data frame with two columns `left` and `right`:

- `left` contains
 - NA for left-censored ($< b$),
 - the lower interval bound for interval-censored,
 - the observed value for exact or right-censored.
- `right` contains
 - NA for right-censored ($> a$),
 - the upper interval bound for interval-censored,
 - the observed value for exact or left-censored.

Thus the pair `(left,right)` encodes:

(a, a) : exact, (a, NA) : right-censored at a , (NA, b) : left-censored at b , (a, b) : interval-censored on (a, b) .

Example. From the `CASdatasets` package we take twenty exit ages and a death indicator:

```

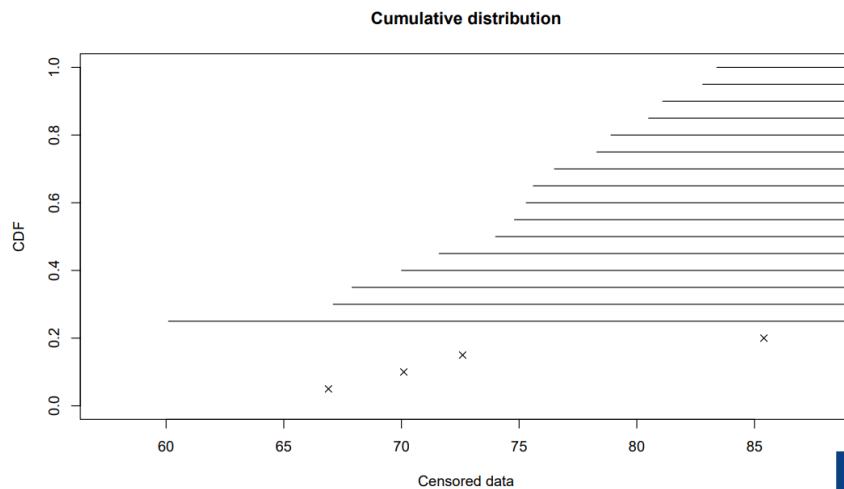
exitage <- c(81.1, 78.9, 72.6, 67.9, 60.1, 78.3, 83.4, 66.9, 74.8, 80.5,
           75.6, 67.1, 75.3, 82.8, 70.1, 85.4, 74, 70, 71.6, 76.5)
death   <- c(0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0)
casdata <- data.frame(left = exitage, right = exitage)
casdata$right[death == 0] <- NA # code nondeaths as rightcensored
print(head(casdata))

```

The resulting table has `right = NA` for censored observations and the exit age in both columns for exact events.

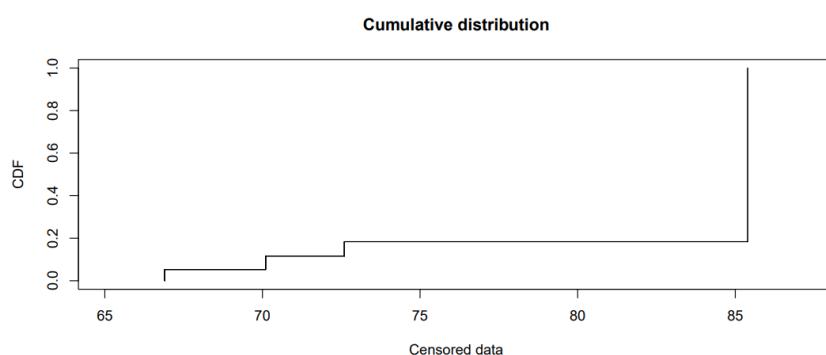
Censored data can be plotted raw

```
fitdistrplus::plotdistcens(casdata, NPMLE = FALSE)
```



or as an empirical distribution

```
fitdistrplus::plotdistcens(casdata)
```

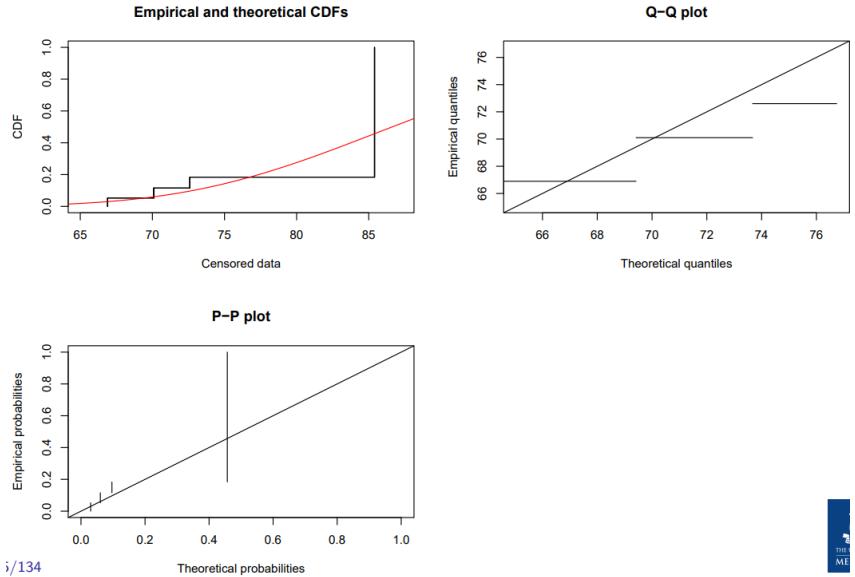


3.7.3 Fitting censored data (in R)

```
cas.gamma.fit <- fitdistrplus::fitdistcens(casdata, "gamma")
summary(cas.gamma.fit)

## Fitting of the distribution ' gamma ' By maximum likelihood on censored## Parameters
##      estimate Std. Error
## shape  57.7678254 41.0552026
## rate   0.6629993  0.5059993
## Loglikelihood: -20.0179 AIC: 44.0358 BIC: 46.02727
## Correlation matrix:
## shape rate
## shape 1.0000000 0.9983085
## rate   0.9983085 1.0000000

plot(cas.gamma.fit)
```



3.7.4 Fitting left truncated data (in R)

In actuarial work we often need to handle data that are recorded only if they lie in a window $(\ell, u]$. If X has pdf $f_X(x)$ and cdf $F_X(x)$, then the ℓ -left-truncated, u -right-truncated variable

$$Y = X \{ \ell < X \leq u \}$$

has density

$$f_Y(y) = \begin{cases} \frac{f_X(y)}{F_X(u) - F_X(\ell)}, & \ell < y \leq u, \\ 0, & \text{otherwise.} \end{cases}$$

Equivalently, its cdf is

$$F_Y(y) = \frac{F_X(y) - F_X(\ell)}{F_X(u) - F_X(\ell)}, \quad \ell < y \leq u.$$

Comments:

- There is no built-in R function for left-truncation; one must adjust the likelihood manually by conditioning on $X > \ell$.
- In the special cases $\ell = -\infty$ or $u = +\infty$ this reduces to simple one-sided truncation.
- For interval-censoring one uses the two-column (`left`, `right`) coding shown earlier.

As an example in R, the `d` and `p` functions of a truncated exponential can be coded as:

```

dtxp <- function(x, rate, low, upp) {
  PU <- pexp(upp, rate = rate)
  PL <- pexp(low, rate = rate)
  dexp(x, rate)/(PU - PL) * (x >= low) * (x <= upp)
}

ptxp <- function(q, rate, low, upp) {
  PU <- pexp(upp, rate = rate)
  PL <- pexp(low, rate = rate)
  (pexp(q, rate) - PL)/(PU - PL) * (q >= low) * (q <= upp) + 1 * (q >
  upp)
}

```

If we generate 200 such truncated variables:

```

set.seed(17032025)
n <- 200 # number of observations
x <- rexp(n) # simulating sample of x's
y <- x[x > 0.5 & x < 3] # truncating to get sample of y's

```

and then fit them with either left- and right- truncation estimated from the data:

```

fit.texp.emp <- fitdist(y, "texp", method = "mle", start = list(rate = 3),
fix.arg = list(low = min(y), upp = max(y)))

# OR if you know the lower and upper bounds you can just manually replace it with
# min(y) and max(y).

```

3.7.5 When truncation and censoring levels are the same everywhere

- in R, the approach will be to code a left-truncated function, and then use the `foocens` functions.
- Let us do this on a gamma distribution:

```

dtgamma <- function(x, rate, shape, low) {
  PL <- pgamma(low, rate = rate, shape = shape)
  dgamma(x, rate = rate, shape = shape)/(1 - PL) * (x >= low)
}

ptgamma <- function(q, rate, shape, low) {
  PL <- pgamma(low, rate = rate, shape = shape)
  (pgamma(q, rate = rate, shape = shape) - PL)/(1 - PL) * (q >= low)
}

```

For instance, assume that truncated at 2 and censored at 20 for all data points.

```

set.seed(22042021)
n <- 2000 # number of observations
x <- rgamma(n, shape = 2, rate = 0.2) # simulating sample of x's
x <- x[x > 2] # left-truncation at 2
n - length(x) # number of observations that were truncated
## [1] 123

censoring <- x > 20 # we will censor at 20
x[x > 20] <- 20 # censoring at 20

# Transform into right format
censoring[censoring == FALSE] <- x[censoring == FALSE]
censoring[censoring == TRUE] <- NA
xcens <- cbind.data.frame(left = x, right = censoring)

# And fitting (NOT alloowing for truncation):
fit.gamma.xcens <- fitdistcens(xcens, "gamma", start = list(
  shape = mean(xcens$left)^2/var(xcens$left),
  rate = mean(xcens$left)/var(xcens$left)))
summary(fit.gamma.xcens)
## Fitting of the distribution ' gamma ' By maximum likelihood on censored## Parameters
##      estimate Std. Error
## shape 2.7153083 0.089141066
## rate  0.2566683 0.009500217
## Loglikelihood: -5412.521 AIC: 10829.04 BIC: 10840.12

# And fitting (ALLOWING for truncation):

```

```

fit.tgamma.xcens <- fitdistcens(xcens, "tgamma", start = list
                                shape = mean(xcens$left)^2/var(xcens$left),
                                rate = mean(xcens$left)/var(xcens$left)),
                                fix.arg = list(low = min(xcens$left)))
summary(fit.tgamma.xcens)
## Fitting of the distribution ' tgamma ' By maximum likelihood on censored## Parameters
##      estimate Std. Error
## shape 2.0296237 0.10529978
## rate  0.2006534 0.01009728
## Fixed parameters:
##      value
## low 2.013575
## Loglikelihood: -5340.151 AIC: 10684.3 BIC: 10695.38

fitdistrplus::cdfcompcens(list(fit.gamma.xcens, fit.tgamma.xcens),
legendtext = c("Not allowing for truncation", "Allowing for truncation"))

```

3.7.6 When truncation and censoring levels vary

In real life, an insurance product would have more than one level of deductibles and limits to suit different policyholders. Simulating another dataset:

```

set.seed(2021)
n <- 3006 # number of observations 9 * 334
orig_x <- rgamma(n, shape = 2, rate = 0.2) # simulating sample of x's
deductibles = rep(c(rep(1, 3), rep(3, 3), rep(5, 3)), 334)
limits = rep(c(15, 20, 30), 3 * 334) + deductibles
# limit is on payment, not raw loss

# MANUALLY APPLYING THE DEDUCTIBLES AND LIMITS
x <- orig_x
censored <- x > limits # we will censor at the limits
x[censored] <- limits[censored] # censoring at the limits
# the above takes only elements of x which have TRUE in the vector
# censored
deducted <- x > deductibles
x <- x[deducted] # left-truncation at all points
# here the truncated observations disappear!
n - length(x) # observations that were truncated
## [1] 378
# that many were removed

claims <- data.frame(x = x, deduct = deductibles[deducted],
                      limitI = censored[deducted])

# PRELIMINARY ANALYSIS
# randomising data, we pretend we do not know how the data was
# generated
claims <- claims[sample(1:nrow(claims), nrow(claims)), ]

hist(claims$x, breaks = 100)
# Note these includes right-censored observations but not the truncated
# values

# PREPARING OUR JOINT LOG LIKELIHOOD FUNCTION
# Here, we are minimising a negative log-likelihood instead of maximising a
# log-likelihood.

```

```

negloglik <- function(pdf, cdf, param, x, deduct, limitI) {
  # Function returns the negative log likelihood of the censored and
  # truncated dataset. Each data point's contribution to the log
  # likelihood depends on the theoretical distribution pdf and cdf
  # and also the deductible and limit values to adjust for truncation
  # and censoring

  PL <- do.call(cdf, c(list(q = deduct), param))
  PX <- do.call(cdf, c(list(q = x), param))
  fX <- do.call(pdf, c(list(x = x), param))

  lik.contr <- ifelse(limitI, log(1 - PX), log(fX)) - log(1 - PL)

  return(-sum(lik.contr))
}

# FITTING THE DISTRIBUTION
# Lets try gamma. Note that our objective function needs starting values for
# the optimisation. What other starting values could we use?

pdf <- dgamma
cdf <- pgamma
x <- claims$x
deduct <- claims$deduct
limitI <- claims$limitI
# MME for starting values
start <- list(shape = mean(x)^2/var(x), rate = mean(x)/var(x))
obj.fun <- function(shape, rate) {
  param <- list(shape = shape, rate = rate)
  return(negloglik(pdf, cdf, param, x, deduct, limitI))
} # we now have a function to minimise wrt shape and rate

gamma.ll.fit <- stats4::mle(obj.fun, start = start, lower = c(0, 0))
summary(gamma.ll.fit)
## Maximum likelihood estimation
##
## Call:
## stats4::mle(minuslogl = obj.fun, start = start, lower = c(0,0))
##
## Coefficients:
##             Estimate Std. Error
## shape  1.9867904  0.084680435
## rate   0.1963044  0.007726585
##
## -2 log L: 15140.45

param.g.ll <- stats4::coef(gamma.ll.fit)
param.g.ll
##      shape      rate
## 1.9867904 0.1963044

# TESTING HOW WELL WE WENT
fit.tcens.param <- param.g.ll # from the proper fit
fit.param <- coef(fitdist(claims$x, "gamma", method = "mle"))
# this is a naive fit

sim.tcens.gamma <- rgamma(10000, shape = fit.tcens.param[1],
                           rate = fit.tcens.param[2]) # sample from proper fit

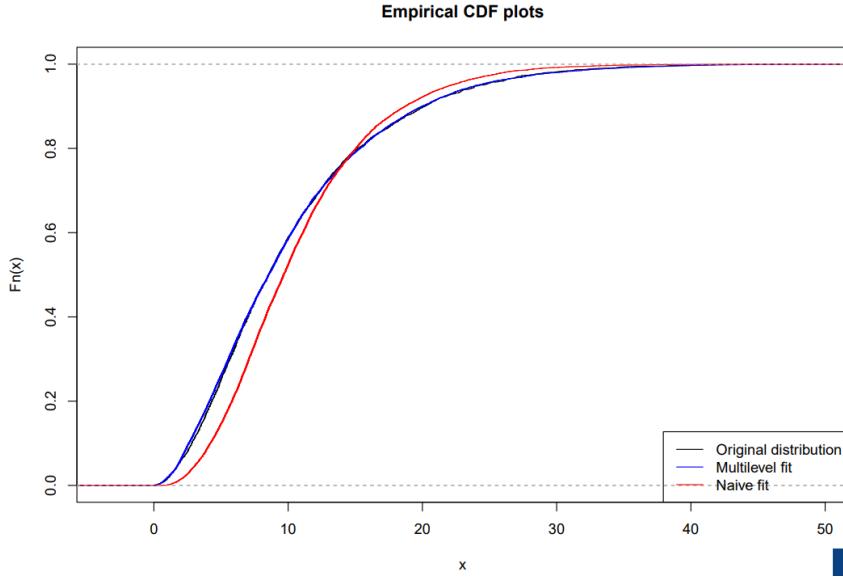
```

```

sim.gamma <- rgamma(10000, shape = fit.param[1], rate = fit.param[2])
# sample from naive fit

# Comparing the proper fit (that accounts for l-trunc and r-cens) with
# a 'naive' fit (that does not account for those)
plot(ecdf(orig_x), main = "Empirical CDF plots", col = "black")
lines(ecdf(sim.tcens.gamma), col = "blue", type = "s")
lines(ecdf(sim.gamma), col = "red", type = "s")
legend("bottomright", legend = c("Original distribution", "Multilevel fit", "Naive fit"),
lty = 1, col = c("black", "blue", "red"))

```



3.8 Model Selection

3.8.1 Graphical Tool Diagnostics

To assess the adequacy of a fitted parametric loss-size model $G(x; \hat{\theta})$, it is standard to use a suite of graphical checks:

1. **Histogram vs. fitted density.** Overlay the histogram of the raw data $\{y_i\}$ with the curve of the fitted density

$$\hat{g}(x) = \frac{\partial}{\partial x} G(x; \hat{\theta}).$$

Look for major shape discrepancies (e.g. peak locations, tail thickness).

2. **Empirical CDF vs. fitted CDF.** Plot the empirical CDF

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{y_i \leq x\}$$

against the theoretical CDF $G(x; \hat{\theta})$. Deviations from the 45° line indicate regions of poor fit.

3. **P–P plot.** For each ordered data point $y_{(i)}$, plot

$$(\hat{F}_n(y_{(i)}), G(y_{(i)}; \hat{\theta})).$$

Points should lie close to the diagonal if the model is correct.

4. **Q–Q plot.** Let $u_i = (i - 0.5)/n$. Plot the sample quantiles $y_{(i)}$ against the theoretical quantiles

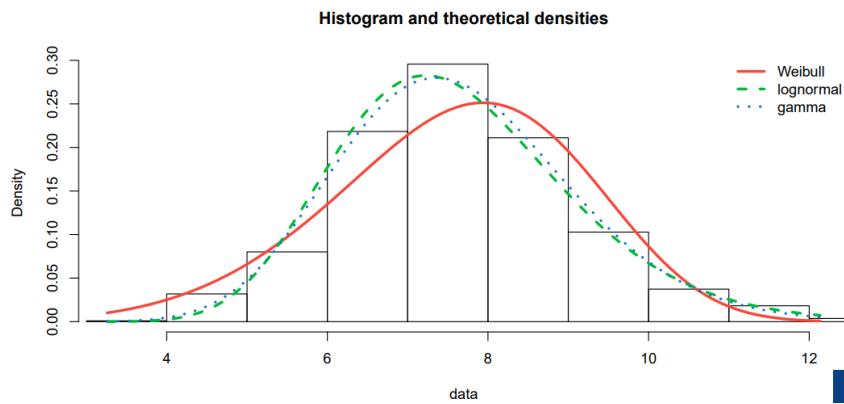
$$G^{-1}(u_i; \hat{\theta}).$$

A straight line indicates good agreement; curvature reveals systematic lack of fit in the center or tails.

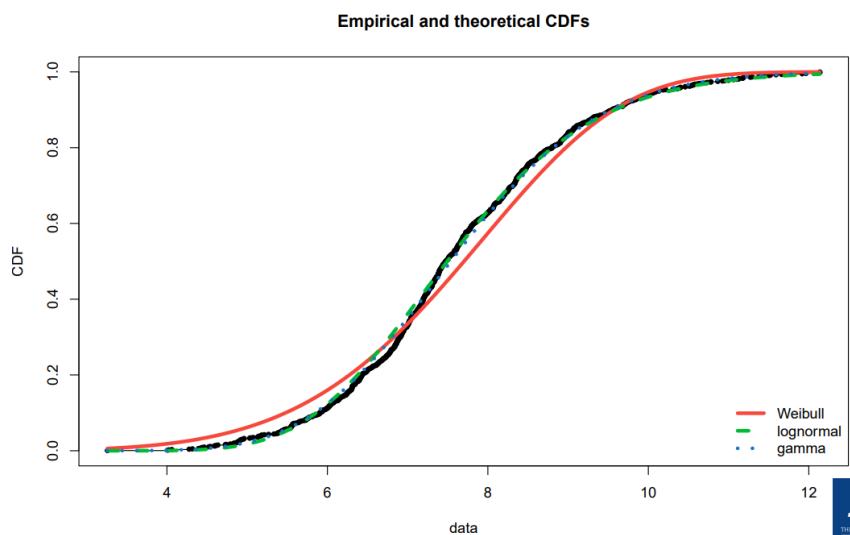
Together these plots provide a quick visual check for features such as skewness, tail weight, and multimodality that the model $G(x; \hat{\theta})$ may fail to capture.

3.8.2 SUVA data Goodness of Fit Plots

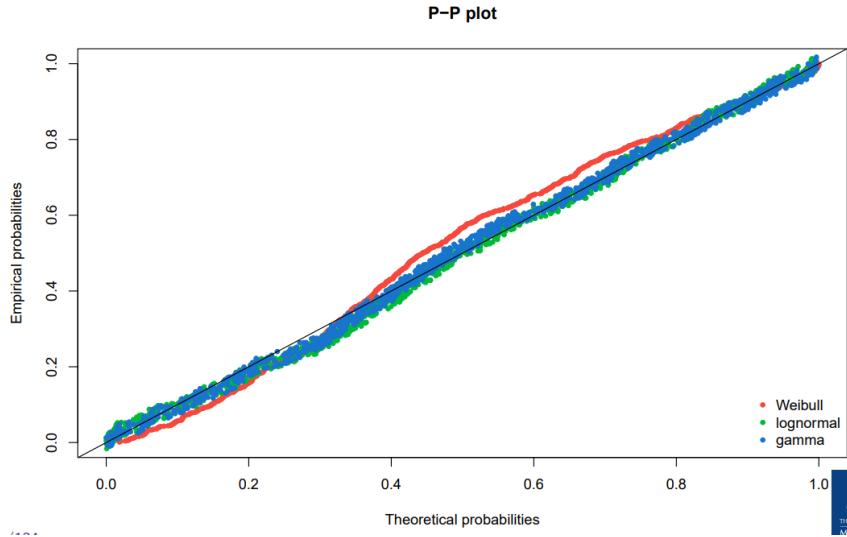
```
plot.legend <- c("Weibull", "lognormal", "gamma")
fitdistrplus::denscomp(list(fit.weibull.mle, fit.lnorm.mle, fit.gamma.mle),
  legendtext = plot.legend, fitlwd = 3)
```



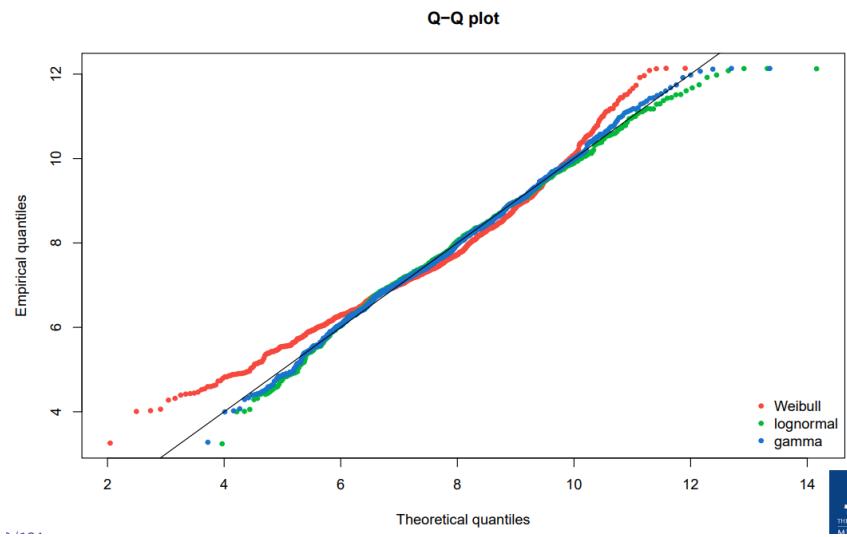
```
fitdistrplus::cdfcomp(list(fit.weibull.mle, fit.lnorm.mle,
  fit.gamma.mle), legendtext = plot.legend, fitlwd = 4, datapch = 20)
```



```
fitdistrplus::ppcomp(list(fit.weibull.mle, fit.lnorm.mle,
  fit.gamma.mle), legendtext = plot.legend, fitpch = 20)
```



```
fitdistrplus::qqcomp(list(fit.weibull.mle, fit.lnorm.mle,
                           fit.gamma.mle), legendtext = plot.legend, fitpch = 20)
```



3.8.3 Hypothesis Tests

3.8.4 Hypothesis Tests for Goodness-of-Fit

We test

$$H_0 : \text{the data come from the fitted model } G(x; \hat{\theta}), \quad H_a : \text{they do not.}$$

Common test statistics are:

- **Kolmogorov–Smirnov (K–S):**

$$D = \sup_x |\hat{G}_n(x) - G(x; \hat{\theta})|.$$

- **Anderson–Darling (A–D):**

$$A^2 = n \int_{-\infty}^{\infty} \frac{[\hat{G}_n(x) - G(x; \hat{\theta})]^2}{G(x; \hat{\theta})[1 - G(x; \hat{\theta})]} dG(x; \hat{\theta}).$$

- χ^2 goodness-of-fit:

$$\chi^2 = \sum_j \frac{(\text{observed}_j - \text{expected}_j)^2}{\text{expected}_j},$$

where the data are grouped into bins.

3.8.5 Using the Test Statistics

- Among candidate models, prefer the one with
 - smallest K–S statistic,
 - smallest A–D statistic,
 - smallest χ^2 (or equivalently largest p -value),
 - largest maximized log–likelihood $\ell(\hat{\theta})$.
- Note: K–S and A–D do *not* adjust for the number of parameters; more complex models tend to fit better by these measures alone.
- The χ^2 test does adjust its degrees of freedom for the number of estimated parameters, but its results can be sensitive to the choice of binning.

3.8.6 Model Selection via Information Criteria

Information criteria penalize model complexity by adding a penalty to the (negative) log–likelihood.

- Akaike Information Criterion (AIC):

$$\text{AIC} = -2 \ell(\hat{\theta}) + 2 d,$$

where d is the number of fitted parameters.

- Bayesian Information Criterion (BIC):

$$\text{BIC} = -2 \ell(\hat{\theta}) + \log(n) d,$$

where n is the sample size.

Smaller AIC or BIC values indicate a preferred model. As $n \rightarrow \infty$, BIC tends to favor simpler models than AIC.

3.8.7 SUVA data Hypothesis Test Statistics

For MLE:

```
gofstat(list(fit.weibull.mle, fit.lnorm.mle, fit.gamma.mle), fitnames = plot.legend)
## Goodness-of-fit statistics
##                               Weibull    lognormal     gamma
## Kolmogorov-Smirnov statistic 0.07105097  0.04276791 0.03385082
## Cramer-von Mises statistic   1.74049707  0.26568341 0.19472283
## Anderson-Darling statistic   10.69572021 1.70209314 1.10574683
##
## Goodness-of-fit criteria
##                               Weibull    lognormal     gamma
## Akaike's Information Criterion 4010.519  3920.717 3907.637
## Bayesian Information Criterion 4020.524  3930.722 3917.641
```

For MME:

```
## Goodness-of-fit statistics
##                               Weibull    lognormal     gamma
## Kolmogorov-Smirnov statistic 0.08023228  0.0374087 0.0327241
## Cramer-von Mises statistic   1.66608962  0.1869752 0.1823197
```

```

## Anderson-Darling statistic 10.53097690 1.4369047 1.0539353
##
## Goodness-of-fit criteria
##                               Weibull lognormal gamma
## Akaike's Information Criterion 4044.239 3922.121 3907.677
## Bayesian Information Criterion 4054.243 3932.125 3917.681

```

R can also provide the results from the GOF hypothesis tests. For example:

```

gammagof <- gofstat(list(fit.gamma.mle, fit.lnorm.mle), fitnames = c("gamma MLE", "lognormal MLE"),
  chisqbreaks = c(10:20/2))
gammagof$chisqvalue
##      gamma MLE lognormal MLE
## 1.378914e-03 1.690633e-05

gammagof$adtest
## gamma MLE lognormal MLE
## "rejected" "not computed"

gammagof$kstest
## gamma MLE lognormal MLE
## "not rejected" "rejected"

gammagof$chisqtable
## obscounts theo gamma MLE theo lognormal MLE
## <= 5 36 23.98533 19.19169
## <= 5.5 28 39.67114 39.53512
## <= 6 60 72.80478 76.73318
## <= 6.5 110 109.49748 116.38286
## <= 7 130 139.03929 145.08202
## <= 7.5 191 152.62215 154.45787
## <= 8 134 147.60896 144.65495
## <= 8.5 141 127.75620 121.96771
## <= 9 91 100.23486 94.30131
## <= 9.5 62 72.06145 67.84850
## <= 10 51 47.90581 45.97086
## > 10 65 65.81255 72.87394

```

3.9 Calculating Within Layers of Claim Size

3.9.1 Deductibles and Policy Limits

Let X be the *ground-up* loss. We introduce two main features:

- **Deductible d :** the insurer only pays the excess above d . Define

$$Y = \begin{cases} 0, & X \leq d, \\ X - d, & X > d, \end{cases} \implies Y = (X - d)_+ = \max(X - d, 0).$$

- **Policy limit M :** the insurer pays up to M . Define

$$Y = \begin{cases} X, & X \leq M, \\ M, & X > M, \end{cases} \implies Y = X \wedge M = \min(X, M).$$

Combining both a deductible d and a limit M , the insurer's payment is

$$Y = \min\{(X - d)_+, M\} = \min\{X \wedge (d + M)\} - (X \wedge d).$$

3.9.2 Useful Formulas for Truncated Moments

First moment of the capped loss.

$$\mathbb{E}[X \wedge M] = \int_0^M x f_X(x) dx + M[1 - F_X(M)] \quad (\text{continuous } X),$$

$$\mathbb{E}[X \wedge M] = \sum_{i:x_i \leq M} x_i P(X = x_i) + M[1 - F_X(M)] \quad (\text{discrete } X).$$

Equivalently, using the survival function $S_X(x) = 1 - F_X(x)$:

$$\mathbb{E}[X \wedge M] = \int_0^M S_X(x) dx, \quad \mathbb{E}[X \wedge M] = \sum_{i:0 \leq x_i < M} S_X(x_i) (x_{i+1} - x_i).$$

k th moment of the capped loss.

$$\mathbb{E}[(X \wedge M)^k] = \int_0^M x^k f_X(x) dx + M^k [1 - F_X(M)] \quad (\text{continuous}),$$

$$\mathbb{E}[(X \wedge M)^k] = \sum_{i:x_i \leq M} x_i^k P(X = x_i) + M^k [1 - F_X(M)] \quad (\text{discrete}).$$

3.9.3 Reinsurance

Reinsurance is a *risk transfer* from an insurer to a reinsurer: the insurer cedes some of its random risk in exchange for a (deterministic) premium. The part the insurer keeps is called the *retention*.

3.9.4 Types of Reinsurance

- **Proportional**
 - *Quota share*: same fixed proportion for every risk.
 - *Surplus*: insurer retains up to a fixed line per risk; excess is ceded.
- **Non-proportional**
 - *Excess-of-loss*: on each individual loss X_i , reinsurer covers $(X_i - d)_+$.
 - *Stop-loss*: on aggregate loss $S = \sum_{i=1}^N X_i$, reinsurer covers $(S - d)_+$.
- **Alternative Risk Transfers (ART)**: e.g. CAT-bonds, longevity bonds, pandemic bonds.

3.9.5 Proportional Reinsurance

If the retained fraction is $\alpha \in [0, 1]$, then for ground-up loss X :

$$Y = \alpha X, \quad Z = (1 - \alpha) X.$$

Hence

$$\mathbb{E}[Y] = \alpha \mathbb{E}[X], \quad \text{Var}(Y) = \alpha^2 \text{Var}(X), \quad \gamma_Y = \gamma_X.$$

Example. If $X \sim \text{Exp}(\beta)$, then

$$\Pr[Y \leq y] = \Pr[\alpha X \leq y] = \Pr[X \leq y/\alpha] = 1 - e^{-\beta y/\alpha},$$

so $Y \sim \text{Exp}(\beta/\alpha)$.

3.9.6 Non-Proportional Reinsurance

Individual Loss Cover For each loss X_i :

$$Y_i = \min(X_i, d), \quad Z_i = (X_i - d)_+.$$

With an upper limit M , one may write

$$Y_i = \min(X_i, d) + (X_i - d - M)_+, \quad Z_i = \min((X_i - d)_+, M).$$

The insurer retains $\sum_i Y_i$, the reinsurer pays $\sum_i Z_i$.

Aggregate (Stop-Loss) Cover On total loss $S = \sum_{i=1}^N X_i$:

$$Y = \min(S, d), \quad Z = (S - d)_+, \quad P_d = \mathbb{E}[(S - d)_+],$$

where P_d is known as the *stop-loss premium*. With an aggregate limit M :

$$Y = \min(S, d) + (S - d - M)_+, \quad Z = \min((S - d)_+, M).$$

3.9.7 Mixed Truncation/Censoring on Retention

Suppose the underlying ground-up loss D is left-truncated at d and right-censored at u . Then the insurer's payment

$$X = \begin{cases} D - d, & d \leq D < u, \\ u - d, & D \geq u, \end{cases}$$

has mixed density

$$f_X(x) = \begin{cases} \frac{f_D(x+d)}{1 - F_D(d)}, & 0 < x < u - d, \\ \frac{1 - F_D(u)}{1 - F_D(d)}, & x = u - d, \\ 0, & \text{otherwise.} \end{cases}$$

In R you can then call `actuar::coverage(pdf=dgamma, cdf=pgamma, deductible=2, limit=20)` ($d = 2$, $u = 20$) to get the corresponding density ($f_X(x)$) for fitting.

```
set.seed(123)

DEDUCTIBLE <- 2
LIMIT <- 20

n <- 2000 # number of observations
x <- rgamma(n, shape = 2, rate = 0.2) # simulating sample of x's
x <- x[x > DEDUCTIBLE] # left-truncation at 2
n <- length(x) # number of observations that were truncated
## [1] 123
censoring <- x > LIMIT # we will censor at 20 (limit)
x[x > LIMIT] <- LIMIT # censoring at 20 (limit)
# Transform into right format
censoring[censoring == FALSE] <- x[censoring == FALSE]
censoring[censoring == TRUE] <- NA
xcens <- cbind.data.frame(left = x, right = censoring)

f <- actuar::coverage(pdf = dgamma, cdf = pgamma, deductible = DEDUCTIBLE, limit = LIMIT)

fit.gamma.xcens2 <- MASS::fitdistr(xcens$left - DEDUCTIBLE, f, start = list(shape =
  mean(xcens$left)^2/var(xcens$left), rate = mean(xcens$left)/var(xcens$left)))
```

```
fit.gamma.xcens2
##      shape          rate
## 2.03990496    0.20142871
## (0.10516140) (0.01009694)

fit.tgamma.xcens # our previous fit with fitdist
## Fitting of the distribution ' tgamma ' on censored data by maximum likelihood
## Parameters:
##      estimate
## shape 2.0296237
## rate 0.2006534
## Fixed parameters:
##      value
## low 2.013575

c(fit.gamma.xcens2$loglik, fit.tgamma.xcens$loglik)
## [1] -5341.551 -5340.151
```

4 Module 4 – Approximations for Compound Distributions

4.1 The $(a, b, 0)$ class of distributions

The (a, b) class of Panjer distributions has the following recursion property:

$$p_k = \Pr[N = k] = \left(a + \frac{b}{k}\right) \Pr[N = k - 1] = \left(a + \frac{b}{k}\right) p_{k-1}, \quad k = 1, 2, \dots$$

or equivalently,

$$\frac{p_k}{p_{k-1}} = a + \frac{b}{k}, \quad k = 1, 2, \dots$$

Hence $\Pr[N = k]$ can be computed recursively from the initial value $p_0 = \Pr[N = 0]$.

Table 2: Members of the $(a, b, 0)$ class

Distribution	a	b	$\Pr[N = 0]$
Poisson (λ)	0	λ	$e^{-\lambda}$
NegBin (γ, p)	$\frac{p}{\gamma}$	$(\gamma - 1)\frac{p}{\gamma}$	$(1-p)^\gamma$
Binomial (ν, p)	$-\frac{p}{1-p}$	$(\nu + 1)\frac{p}{1-p}$	$(1-p)^\nu$

Example: Poisson and Binomial

For the Poisson pmf,

$$\frac{\Pr[N = n]}{\Pr[N = n - 1]} = \frac{e^{-\lambda} \lambda^n / n!}{e^{-\lambda} \lambda^{n-1} / (n-1)!} = \frac{\lambda}{n} = 0 + \frac{\lambda}{n},$$

so

$$a = 0, \quad b = \lambda.$$

For the Binomial pmf,

$$\frac{\Pr[N = n]}{\Pr[N = n - 1]} = \frac{\binom{\nu}{n} p^n (1-p)^{\nu-n}}{\binom{\nu}{n-1} p^{n-1} (1-p)^{\nu-n+1}} = \frac{\nu - n + 1}{n} \frac{p}{1-p} = \frac{p}{1-p} \left(-1 + \frac{\nu + 1}{n}\right) = -\frac{p}{1-p} + \frac{(\nu + 1)p}{1-p} \frac{1}{n},$$

so

$$a = -\frac{p}{1-p}, \quad b = \frac{(\nu + 1)p}{1-p}.$$

4.2 The $(a, b, 1)$ class of distributions

A discrete random variable is a member of the $(a, b, 1)$ class of distributions if there exist constants a and b such that

$$\frac{p_k}{p_{k-1}} = a + \frac{b}{k}, \quad k = 2, 3, \dots$$

Note:

- The recursion starts at $k = 2$ for the $(a, b, 1)$ class.
- The extra freedom allows the probability at zero to be set to any value $0 \leq p_0 \leq 1$.

4.2.1 Zero-truncated distributions

Setting $p_0 = 0$ in the $(a, b, 1)$ class defines the subclass of **zero-truncated distributions**. Let p_k^T denote the probability mass at k for a zero-truncated distribution. Then

$$p_k^T = \begin{cases} 0, & k = 0, \\ \frac{p_k}{1 - p_0}, & k = 1, 2, \dots, \end{cases}$$

where p_k is the pmf of the corresponding member of the $(a, b, 0)$ class.

Members include:

- zero-truncated Poisson (`actuar::ztpois`),
- zero-truncated binomial (`actuar::ztnbinom`),
- zero-truncated negative binomial (`actuar::ztnbinom`),
- zero-truncated geometric (`actuar::ztgeom`).

4.2.2 Zero-modified distributions

Setting $p_0 \equiv p_0^M$ ($0 < p_0^M < 1$) in the $(a, b, 1)$ class defines the subclass of **zero-modified distributions**. These distributions are discrete mixtures between a degenerate mass at zero and the corresponding $(a, b, 0)$ member. Let p_k^M denote the probability mass at k for a zero-modified distribution. Then

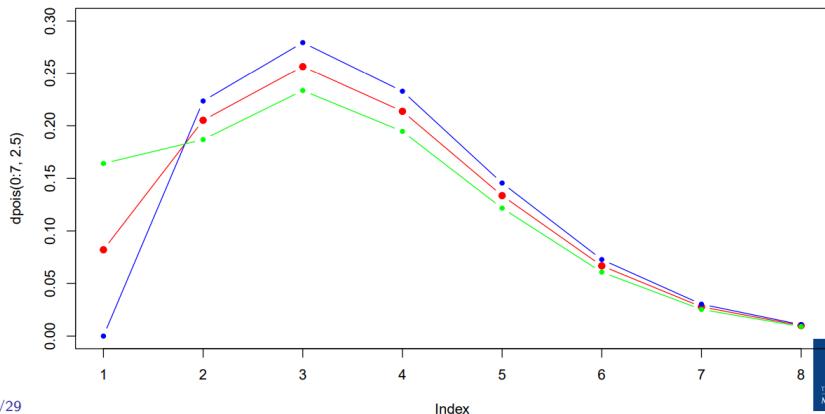
$$p_k^M = \begin{cases} p_0^M, & k = 0, \\ \frac{1 - p_0^M}{1 - p_0} p_k, & k = 1, 2, \dots, \end{cases}$$

where p_k is the pmf of the corresponding member of the $(a, b, 0)$ class.

Members include the zero-modified

- Poisson (`actuar::zmpois`),
- Binomial (`actuar::zmbinom`),
- Negative binomial (`actuar::zmnb`),
- Geometric (`actuar::zmgeom`).

```
plot(dpois(0:7, 2.5), pch = 20, col = "red", ylim = c(0, 0.3),
cex = 1.5, type = "b")
points(dztpois(0:7, 2.5), pch = 20, col = "blue", type = "b")
points(dzmpois(0:7, 2.5, 2 * dpois(0, 2.5)), pch = 20, col = "green",
type = "b")
```



4.3 Panjer's recursion algorithm

- The remarkable property of the (a, b) class of frequency distributions allows us to develop a *recursive* method to get the distribution of S for discrete Y 's.
- When Y is continuous, simply discretise its cdf first.

Let S have a compound distribution on Y , i.e. $S = \sum_{i=1}^N Y_i$, where the following are mutually independent:

- N belongs to the (a, b) class of distributions;
- the Y_i are i.i.d., non-negative and discrete with pmf $g_j = \Pr(Y = j)$.

Then for $s = 1, 2, \dots$,

$$f_S(s) = \frac{1}{1 - a g_0} \sum_{j=1}^s \left(a + \frac{b j}{s} \right) g_j f_S(s-j),$$

with starting value

$$f_S(0) = \begin{cases} \Pr(N = 0), & g_0 = 0, \\ M_N(\ln g_0), & g_0 > 0, \end{cases}$$

where $M_N(t) = E[e^{tN}]$ is the mgf of N . Note that if $g_y = 0$ for $y > y_{\max}$, the upper limit of the sum may be replaced by $\min(s, y_{\max})$.

Example: Panjer's recursion for compound Poisson

If $S \sim \text{compound Poisson}(\lambda, \{g_j\})$, then N is Poisson with parameters $a = 0, b = \lambda$ so Panjer's formula gives for $s = 1, 2, \dots$

$$f_S(s) = \frac{\lambda}{s} \sum_{j=1}^s j g_j f_S(s-j),$$

with starting value (whether g_0 is zero or not)

$$f_S(0) = \exp[\lambda(g_0 - 1)].$$

Example 12.4.2 (Bowers et al., 1997)

Let $N \sim (\lambda = 0.8)$, and let claim sizes Y_i be i.i.d. with

$$\Pr(Y = 1) = 0.25, \quad \Pr(Y = 2) = 0.375, \quad \Pr(Y = 3) = 0.375.$$

Define the aggregate loss

$$S = \sum_{i=1}^N Y_i.$$

Use Panjer's recursion to compute $f_S(s) = \Pr(S = s)$ for $s = 0, 1, 2, 3, \dots$

Solution. Here $a = 0, b = \lambda = 0.8$, and $g_j = \Pr(Y = j)$. Panjer's recursion gives

$$f_S(s) = \frac{\lambda}{s} \sum_{j=1}^s j g_j f_S(s-j), \quad s = 1, 2, \dots,$$

with starting value

$$f_S(0) = \Pr(N = 0) = e^{-0.8} = 0.44933.$$

Hence

$$f_S(1) = \frac{0.8}{1} (1 \cdot 0.25 f_S(0)) = 0.2 e^{-0.8} = 0.089866,$$

$$f_S(2) = \frac{0.8}{2} (1 \cdot 0.25 f_S(1) + 2 \cdot 0.375 f_S(0)) = 0.32 e^{-0.8} = 0.14379,$$

$$f_S(3) = \frac{0.8}{3} (1 \cdot 0.25 f_S(2) + 2 \cdot 0.375 f_S(1) + 3 \cdot 0.375 f_S(0)) = 0.3613 e^{-0.8} = 0.16236,$$

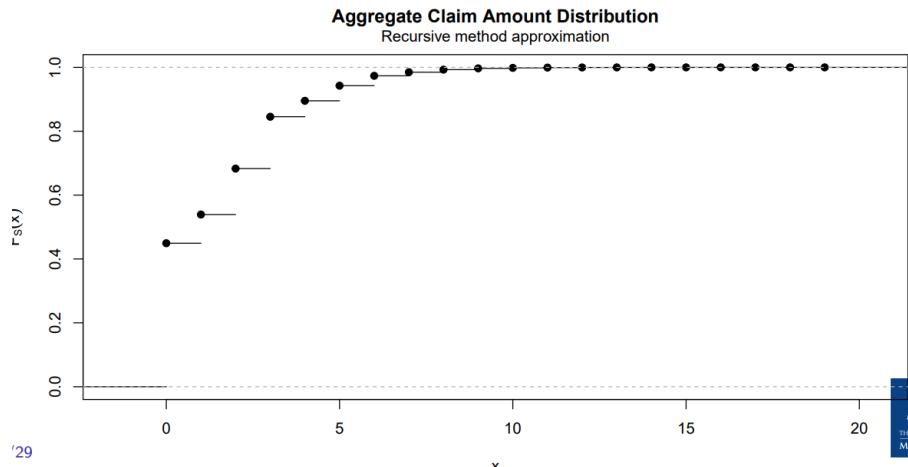
etc.

In R:

```
fs <- actuar::aggregateDist(method = "recursive", model.freq = "poisson",
lambda = 0.8, model.sev = c(0, 0.25, 0.375, 0.375), x.scale = 1)
diff(fs)[1:4]
fs(knots(fs)) # supposed to return cdf, haven't tested but is in documentation

## [1] 0.44932896 0.08986579 0.14378527 0.16235753

plot(fs)
```



4.3.1 Panjer's recursion in R

Panjer's recursion can be performed using the `aggregateDist` function with `method="recursive"`.

- The frequency distribution can be any member of the $(a, b, 0)$ or $(a, b, 1)$ class (i.e. with arbitrary mass at zero).
- The severity distribution must be discrete on $\{0, 1, \dots, m\}$ for some monetary unit.

Important parameters include:

- `model.freq`: name of the frequency distribution (e.g. `"poisson"`, `"nbinom"`, etc.).
- `model.sev`: data frame or named vector giving the pmf of the discretized severity distribution Y .
- `x.scale`: the value of one monetary unit in the severity model (i.e. the increment size of the support).
- `diff(model)`: returns a vector of the pmf values.
- `model(knots(model))`: returns a vector of cdf evaluated at knots.

4.4 Approximations

Possible motivations for using approximations to the compound distribution S :

- Exact computation of the distribution of S may be infeasible:
 - no detailed data beyond the moments of S ,
 - technical difficulties in fitting a tractable model.

- A sophisticated model may be wrong (limited data, etc.) and an approximation can be sufficiently accurate.
- Quick results are needed, or high precision is not justified by available resources.

Notation: In this section we write

$$s_S \equiv \gamma_1(S), \quad \gamma_2(S) \equiv \gamma_2.$$

4.4.1 Normal approximation

By the Central Limit Theorem,

$$\Pr[S \leq s] = \Pr\left[\frac{s - \mathbb{E}[S]}{\sqrt{(S)}} \leq Z\right] \approx \Phi\left(\frac{s - \mathbb{E}[S]}{\sqrt{(S)}}\right),$$

where $Z \sim N(0, 1)$ and Φ is its cdf.

- The classical CLT holds for a fixed number of summands, whereas here N is random.
- In the special case $S \sim \text{CompPoi}(\lambda\nu, G)$, Theorem 4.1 of Wüthrich (2024) shows

$$\frac{S - \lambda\nu \mathbb{E}[Y]}{\sqrt{\lambda\nu \mathbb{E}[Y^2]}} \implies N(0, 1) \quad (\nu \rightarrow \infty),$$

leading to

$$\Pr[S \leq s] \approx \Phi\left(\frac{s - \lambda\nu \mathbb{E}[Y]}{\sqrt{\lambda\nu \mathbb{E}[Y^2]}}\right).$$

This requires $\mathbb{E}[Y^2] < \infty$, so is best used for S_{sc} .

4.4.2 Translated Gamma and Log-Normal approximations

Idea: Instead of correcting the CLT, choose a positively skewed distribution and match its first three moments via a location parameter k . Let

$$X = k + Z, \quad Z \sim \begin{cases} \Gamma(\gamma, c), \\ \text{Lognormal}(\mu, \sigma^2). \end{cases}$$

Match $\mathbb{E}[X] = \mathbb{E}[S]$, $(X) = (S)$, and skewness $\gamma_1(X) = s_S$. Here k only enters the mean.

Fitting a translated gamma to $S \sim \text{CompPoi}(\lambda\nu, G)$: We solve

$$\begin{cases} \mathbb{E}[S] = \lambda\nu \mathbb{E}[Y] = k + \gamma/c, \\ (S) = \lambda\nu \mathbb{E}[Y^2] = \gamma/c^2, \\ s_S = \frac{\mathbb{E}[Y^3]}{(\lambda\nu)^{1/2} \mathbb{E}[Y^2]^{3/2}} = 2\gamma^{-1/2}. \end{cases}$$

Hence

$$\gamma = \left(\frac{2(\lambda\nu)^{1/2} \mathbb{E}[Y^2]^{3/2}}{\mathbb{E}[Y^3]}\right)^2, \quad c = \left(\frac{\gamma}{\lambda\nu \mathbb{E}[Y^2]}\right)^{1/2}, \quad k = \lambda\nu \mathbb{E}[Y] - \frac{\gamma}{c}.$$

Example: Normal and translated Gamma approximation

Suppose $S \sim \text{CompPoi}(\lambda = 16, Y \equiv 1)$. Find $\Pr(S \leq 25)$ by both methods.

Normal approximation

$$\Pr(S \leq 25) \approx \Phi\left(\frac{25 - 16}{\sqrt{16}}\right) = \Phi(2.25) = 0.98776.$$

Translated Gamma approximation Here $\nu = 1$ and $Y \equiv 1 \implies \mathbb{E}[Y] = 1, \mathbb{E}[Y^2] = 1, \mathbb{E}[Y^3] = 1$. The moment-matching equations become

$$\frac{\gamma}{c} + k = 16, \quad \frac{\gamma}{c^2} = 16, \quad 2/\sqrt{\gamma} = 1/4,$$

whence $\gamma = 64, c = 2, k = -16$. Thus

$$\Pr(S \leq 25) \approx \Pr(X < 25) = \Pr(Z < 25 - k) = \Pr(Z < 41), \quad Z \sim \Gamma(64, 2),$$

$$\Pr(Z < 41) = \frac{2^{64}}{\Gamma(64)} \int_0^{41} x^{63} e^{-2x} dx = \text{pgamma}(41, 64, 2) = 0.98252.$$

5 Module 5 - Copulas

5.1 Measures of dependence

Independence \implies zero correlation BUT zero correlation $\not\Rightarrow$ independence.

5.1.1 Pearson's correlation measure

Pearson's correlation coefficient is defined by

$$\rho(Z_i, Z_j) = \rho_{ij} = \frac{\text{Cov}(Z_i, Z_j)}{\sqrt{\text{Var}(Z_i)\text{Var}(Z_j)}}$$

Note:

- this measures the degree of **linear relationship** between Z_i and Z_j .
- its value is between -1 and 1 .
- in general, it does not reveal all the information of the dependence structure of random couples.

```
cor(data, method = "pearson") # default, data needs to have more than 1 column
```

5.1.2 Kendall's tau

Kendall's tau rank correlation coefficient is defined by

$$\tau(Z_i, Z_j) = \tau_{ij} = \Pr[(Z_i - Z'_i)(Z_j - Z'_j) > 0] - \Pr[(Z_i - Z'_i)(Z_j - Z'_j) < 0]$$

where (Z_i, Z_j) and (Z'_i, Z'_j) are two independent realisations.

Note:

- the first term is called the probability of concordance, the latter, probability of disconcordance.
- its value is also between -1 and 1 .
- it can be shown to equal: $\tau(Z_i, Z_j) = 4\mathbb{E}[F(Z_i, Z_j)] - 1$.
- concordance and disconcordance only depends on **ranks**.

```
cor(data, method = "kendall") # data needs to have more than 1 column  
# Kendall's tau is unchanged under transformations that do not affect rank (eg. log)
```

5.1.3 Spearman's rho

Spearman's rho rank correlation coefficient is defined by

$$r(Z_i, Z_j) = r_{ij} = \rho(F_i(Z_i), F_j(Z_j))$$

where F_i and F_j are the respective marginal distributions. Note:

- it is a measure of Pearson's correlation but applied to the transformed variables.
- its value is also between -1 and 1 .
- it is directly formulated on ranks.

```
cor(data, method = "spearman") # data needs to have more than 1 column  
# Kendall's tau is unchanged under transformations that do not affect rank (eg. log)
```

5.2 Copula Theory

5.2.1 Sklar's representation theorem

Copula links the joint distribution function to its marginals. Sklar's theorem says there exists a copula function C such that:

$$F(x_1, x_2, \dots, x_n) = C(F_1(x_1), F_2(x_2), \dots, F_n(x_n))$$

where F_k is the marginal df of X_k , $k = 1, 2, \dots, n$. Equivalently,

$$\Pr(X_1 \leq x_1, \dots, X_n \leq x_n) = C(\Pr[X_1 \leq x_1], \dots, \Pr[X_n \leq x_n])$$

An n -dimensional copula is a df on $[0, 1]^n$ with standard uniform marginal distributions.

Under certain conditions, the copula:

$$C(u_1, \dots, u_n) = \Pr[U_1 \leq F_1(X_1), \dots, U_n \leq F_n(X_n)] = F(F_1^{-1}(u_1), \dots, F_n^{-1}(u_n))$$

is unique, where F_k^{-1} denote the quantile functions.

5.2.2 When is a Copula valid

For $n = 2$, C is a function mapping $[0, 1]^2$ to $[0, 1]$ that is non-decreasing and right continuous, and:

1. $\lim_{u_k \rightarrow 0} C(u_1, u_2) = 0$ for $k = 1, 2$;
2. $\lim_{u_1 \rightarrow 1} C(u_1, u_2) = u_2$ and $\lim_{u_2 \rightarrow 1} C(u_1, u_2) = u_1$;
3. C satisfies the inequality

$$C(v_1, v_2) - C(u_1, v_2) - C(v_1, u_2) + C(u_1, u_2) \geq 0 \quad \text{for any } u_1 \leq v_1, u_2 \leq v_2.$$

Corresponding heuristics are:

1. If the event on one variable is impossible, then the joint probability is impossible.
2. If the event on one variable is certain, then the joint probability boils down to the marginal of the other one.
3. There cannot be negative probabilities.

5.2.3 Constructing Copulas from df's examples

Example 1: Let

$$F(x, y) = \begin{cases} \frac{(x+1)(e^y - 1)}{x + 2e^y - 1}, & (x, y) \in [-1, 1] \times [0, \infty), \\ 1 - e^{-y}, & (x, y) \in (1, \infty) \times [0, \infty), \\ 0, & \text{elsewhere.} \end{cases}$$

Hence

$$F(x, \infty) = F(x) = \frac{x+1}{2} \equiv u, \quad x \in [-1, 1], \quad F^{-1}(u) = 2u - 1 = x,$$

$$F(1, y) = G(y) = 1 - e^{-y} \equiv v, \quad y \geq 0, \quad G^{-1}(v) = -\ln(1-v) = y.$$

Finally,

$$\begin{aligned} C(u, v) &= \frac{(2u-1+1)[(1-v)^{-1}-1]}{2u-1+2((1-v)^{-1}-1)} = \frac{2u(1-1+v)}{(2u-2)(1-v)+2} \\ &= \frac{2uv}{2u-2uv-2+2v+2} = \frac{uv}{u+v-uv} = uv \times \frac{1}{u+v-uv}. \end{aligned}$$

Note:

- Independence copula is $C(u, v) = uv$, here “tweaked” by a function of u and v .
- The copula captures the dependence structure, while separating the effects of the marginals (which are behind probabilities u and v).
- Other copulas generally contain parameter(s) to fine-tune the strength of dependence.

Example 2: Suppose (X_1, X_2) has a bivariate distribution function described by

$$F(X_1 \leq x_1, X_2 \leq x_2) = \exp\left\{-\left[\left(\log \frac{1}{1-e^{-x_1}}\right)^\alpha + \left(\log \frac{1}{1-e^{-x_2}}\right)^\alpha\right]^{1/\alpha}\right\}, \quad \alpha \geq 1.$$

Derive the marginal distributions of X_1 and X_2 and express the copula function C such that

$$F(X_1 \leq x_1, X_2 \leq x_2) = C(F_{X_1}(x_1), F_{X_2}(x_2)) = C(u, v).$$

By letting $x_2 \rightarrow \infty$, we find the marginal of X_1 :

$$F_{X_1}(x_1) = F(x_1, \infty) = 1 - e^{-x_1} = u.$$

Similarly, by letting $x_1 \rightarrow \infty$, we find the marginal of X_2 :

$$F_{X_2}(x_2) = F(\infty, x_2) = 1 - e^{-x_2} = v.$$

Both have $\text{Exp}(1)$ distribution. We then have

$$C(u, v) = F(X_1 \leq x_1, X_2 \leq x_2) = \exp\left\{-\left[(-\log u)^\alpha + (-\log v)^\alpha\right]^{1/\alpha}\right\}.$$

5.2.4 Invariance property of Copulas

- Suppose random vector \mathbf{X} has copula C and suppose T_1, \dots, T_n are strictly increasing functions of X_1, \dots, X_n , respectively.
- The random vector defined by $(T_1(X_1), \dots, T_n(X_n))$ has the same copula C —this is referred to as the **invariance property**.
- The usefulness of this property can be illustrated in many ways. If you have a copula describing joint distribution of insurance losses of various types, and you decide the quantity of interest is a transformation (e.g. logarithm) of these losses, then the multivariate distribution structure does not change.
- The copula is then also invariant to inflation.
- Only the marginal distributions change.

5.2.5 The Fréchet Bounds

Define the Fréchet bounds as:

- **Fréchet lower bound:**

$$L_F(u_1, \dots, u_n) = \max\left(\sum_{k=1}^n u_k - (n-1), 0\right)$$

- **Fréchet upper bound:**

$$U_F(u_1, \dots, u_n) = \min(u_1, \dots, u_n)$$

All copula satisfy the following bounds:

$$L_F(u_1, \dots, u_n) \leq C(u_1, \dots, u_n) \leq U_F(u_1, \dots, u_n).$$

5.2.6 Fundamental copulas

Fundamental copulas contain special dependence structures.

- The independence copula introduced earlier $\Pi(u, v) = uv$ reflects the independence (i.e., lack of dependence) of the rvs.
- The **comonotonicity** copula is the Fréchet upper bound copula with

$$M(u, v) = \min(u, v).$$

- This special copula is the joint df of random vector (V, V) where $V \sim \text{Uniform}(0, 1)$, reflecting the **perfectly positive dependence** structure of the rvs.
- In other words, there is a single source of risk and the comonotonic variables are some increasing transformations of that risk.
- The **countermanotonicity** copula is the Fréchet lower bound copula with

$$W(u, v) = \max(u + v - 1, 0).$$

- This special copula is the joint df of random vector $(V, 1-V)$ where $V \sim \text{Uniform}(0, 1)$, reflecting the **perfectly negative dependence** structure of the rvs.
- The Fréchet upper bound satisfies the definition of a copula, but the Fréchet lower bound does not for dimensions $n \geq 3$. In other words, the countermanotonicity copula only exists for dimension $n = 2$, whereas the comonotonicity copula can be obtained for dimensions $n \geq 2$.

5.2.7 Survival copulas

What if we want to work with survival functions

$$\bar{F}_i(x_i) = 1 - F_i(x_i) = S_i(x_i)$$

rather than distribution functions?

→ We can couple the \bar{F}_i 's with the **survival copula** \bar{C} .

In the bivariate case, this yields

$$\bar{F}(x_1, x_2) = \Pr[X_1 > x_1, X_2 > x_2] = \bar{C}(\bar{F}_1(x_1), \bar{F}_2(x_2)),$$

where

$$\bar{C}(1-u, 1-v) = 1 - u - v + C(u, v).$$

This is because

$$\Pr[X_1 > x_1, X_2 > x_2] = 1 - \Pr[X_1 \leq x_1] - \Pr[X_2 \leq x_2] + \Pr[X_1 \leq x_1, X_2 \leq x_2].$$

5.3 Tail Dependence

5.3.1 Coefficient of Lower Tail Dependence

The coefficient of **lower tail dependence** is defined as

$$\lambda_L = \lim_{u \rightarrow 0^+} \Pr[X_1 \leq F_{X_1}^{-1}(u) \mid X_2 \leq F_{X_2}^{-1}(u)] = \lim_{u \rightarrow 0^+} \frac{C(u, u)}{u},$$

where $\lambda_L \in [0, 1]$.

Examples:

$$\lambda_L^\Pi = \lim_{u \rightarrow 0^+} \frac{u \cdot u}{u} = \lim_{u \rightarrow 0^+} u = 0, \quad \lambda_L^M = \lim_{u \rightarrow 0^+} \frac{\min(u, u)}{u} = \lim_{u \rightarrow 0^+} 1 = 1.$$

5.3.2 Coefficient of Upper Tail Dependence

The coefficient of **upper tail dependence** is defined similarly but using the survival copula, which yields

$$\lambda_U = \lim_{u \rightarrow 1^-} \Pr[X_1 \geq F_{X_1}^{-1}(u) \mid X_2 \geq F_{X_2}^{-1}(u)] = \lim_{u \rightarrow 1^-} \frac{\bar{C}(1-u, 1-u)}{1-u} = \lim_{u \rightarrow 0^+} \frac{\bar{C}(u, u)}{u}.$$

Note $\bar{C}(u, u) = 2u - 1 + C(1-u, 1-u)$ and $\lambda_U \in [0, 1]$.

Examples:

$$\lambda_U^I = \lim_{u \rightarrow 1^-} \frac{1-2u+u^2}{1-u} = \lim_{u \rightarrow 1^-} (1-u) = 0, \quad \lambda_U^M = \lim_{u \rightarrow 1^-} \frac{1-2u+\min(u, u)}{1-u} = \lim_{u \rightarrow 1^-} 1 = 1.$$

5.4 Archimedean copulas

C is **Archimedean** if it has an explicit closed-form

$$C(u_1, \dots, u_n) = \psi^{-1}(\psi(u_1) + \dots + \psi(u_n)),$$

for all $0 \leq u_1, \dots, u_n \leq 1$ and for some continuous function ψ (called the *generator*) satisfying:

1. $\psi(1) = 0$;
2. ψ is strictly decreasing ($\psi'(t) < 0$); and
3. ψ is convex ($\psi''(t) \geq 0$).
4. *This is NOT a requirement*, but $\psi(t)$ maps $[0, 1] \rightarrow [0, \infty]$ and $\psi^{-1}(t)$ maps $[0, \infty] \rightarrow [0, 1]$.

The Kendall's τ and the generator function $\psi(t)$ are related via

$$\tau = 1 + 4 \int_0^1 \frac{\psi(t)}{\psi'(t)} dt \in [-1, 1]$$

Note:

- If $\tau = -1$ it is counter-monotonicity copula since it is perfectly dis-concordant (perfectly negatively correlated).
- If $\tau = 1$ it is co-monotonicity copula since it is perfectly concordant (perfectly positively correlated).

5.4.1 Clayton Copula

The bivariate Clayton copula is defined by

$$C(u_1, u_2) = (u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta}, \quad \theta \in (0, \infty).$$

It is of Archimedean type with:

- $\psi(t) = \frac{1}{\theta}(t^{-\theta} - 1)$
- $\psi^{-1}(s) = (1 + \theta s)^{-1/\theta}$

The relationship between Kendall's τ and θ is

$$\tau = \frac{\theta}{2+\theta} \iff \theta = \frac{2\tau}{1-\tau}.$$

The tail-dependence coefficients are

$$\lambda_L = \lim_{u \rightarrow 0^+} \frac{C(u, u)}{u} = 2^{-1/\theta}, \quad \lambda_U = \lim_{u \rightarrow 1^-} \frac{\bar{C}(1-u, 1-u)}{1-u} = 0.$$

Limiting cases:

- As $\theta \rightarrow 0$,

$$C(u, v) \rightarrow uv,$$

the independence copula.

- As $\theta \rightarrow \infty$,

$$C(u, v) \rightarrow \min(u, v),$$

the comonotonicity copula.

- (Bivariate only) Extending to $\theta \rightarrow (-1)^+$ yields

$$C(u, v) \rightarrow \max(u + v - 1, 0),$$

the countermonotonicity copula.

5.4.2 Frank copula

The bivariate Frank copula is defined by

$$C(u_1, u_2) = -\frac{1}{\theta} \ln \left(1 + \frac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{e^{-\theta} - 1} \right), \quad \theta \in \mathbb{R} \setminus \{0\}.$$

It is of Archimedean type with:

- $\psi(t) = -\ln\left(\frac{e^{-\theta t} - 1}{e^{-\theta} - 1}\right)$,
- $\psi^{-1}(s) = -\frac{1}{\theta} \ln(1 + e^{-s}(e^{-\theta} - 1))$.

The relationship between Kendall's τ and θ is

$$\tau = 1 - \frac{4}{\theta} + \frac{4}{\theta^2} \int_0^\theta \frac{t}{e^t - 1} dt.$$

The tail-dependence coefficients are

$$\lambda_L = \lim_{u \rightarrow 0^+} \frac{C(u, u)}{u} = 0, \quad \lambda_U = \lim_{u \rightarrow 1^-} \frac{\bar{C}(1-u, 1-u)}{1-u} = 0.$$

Limiting cases:

- As $\theta \rightarrow 0$,

$$C(u, v) \rightarrow uv,$$

the independence copula.

- As $\theta \rightarrow +\infty$,

$$C(u, v) \rightarrow \min(u, v),$$

the comonotonicity copula.

- As $\theta \rightarrow -\infty$,

$$C(u, v) \rightarrow \max(u + v - 1, 0),$$

the countermonotonicity copula (bivariate only).

5.4.3 Gumbel Copula

The bivariate Gumbel–Hougaard copula is defined by

$$C(u_1, u_2) = \exp\left(-[(-\ln u_1)^\theta + (-\ln u_2)^\theta]^{1/\theta}\right), \quad \theta \in [1, \infty).$$

It is of Archimedean type with:

- $\psi(t) = (-\ln t)^\theta$,

- $\psi^{-1}(s) = \exp(-s^{1/\theta})$.

The relationship between Kendall's τ and θ is

$$\tau = \frac{\theta - 1}{\theta} \iff \theta = \frac{1}{1 - \tau}.$$

The tail-dependence coefficients are

$$\lambda_L = \lim_{u \rightarrow 0^+} \frac{C(u, u)}{u} = 0, \quad \lambda_U = \lim_{u \rightarrow 1^-} \frac{C(1-u, 1-u)}{1-u} = 2 - 2^{1/\theta}.$$

Limiting cases:

- $\theta = 1 \implies C(u, v) = uv$, the independence copula.
- $\theta \rightarrow \infty \implies C(u, v) \rightarrow \min(u, v)$, the comonotonicity copula.

6 Module 6 – Extreme Value Theory (EVT)

6.1 Measures of tail weight

Models that assign higher probabilities to these large values are said to be heavier-tailed. Tail behaviour can be examined via

1. **log-log plot:** linear decrease for heavy tails
2. **Mean Excess function:** linear increase for heavy tails
3. **Existence of Moments:** Less moments means more heavy tails (e.g. Pareto vs. gamma)
4. **Limiting Density Ratio:** ratio of density function as the right tail of two distributions

$$\lim_{x \rightarrow \infty} \frac{S_A(x)}{S_B(x)} = \lim_{x \rightarrow \infty} \frac{f_A(x)}{f_B(x)}$$

- If limit = c (constant): The tails are comparable in size
 - If limit = ∞ : Distribution A has a heavy tail than B
 - If limit = 0: Distribution A has a thinner tail than B
5. **Hazard Rate function:** $h(x) = \frac{f(x)}{1-F(x)}$, constant for exponential and decreasing for heavy tails.

Example: Exponential vs. Pareto (limit of survival functions)

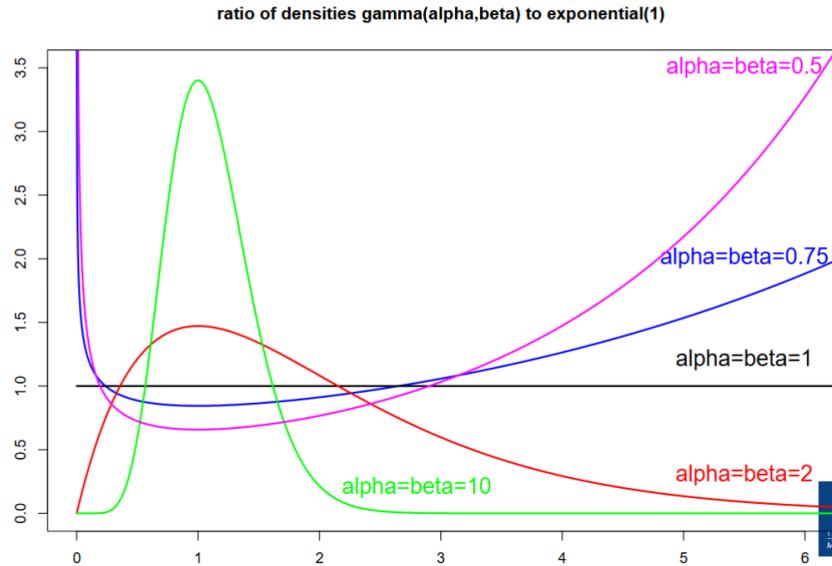
$$\begin{aligned} \lim_{x \rightarrow \infty} \frac{S_{\text{Pareto}}(x)}{S_{\text{Exponential}}(x)} &= \lim_{x \rightarrow \infty} \frac{\frac{\theta^\alpha}{(x+\theta)^\alpha}}{e^{-\theta x}} \\ &= \lim_{x \rightarrow \infty} \frac{\theta^\alpha e^{\theta x}}{(x+\theta)^\alpha} \rightarrow \infty, \text{ so pareto has heavier tail} \end{aligned}$$

Example: Exponential vs. Pareto (hazard rate function)

$$\begin{aligned} h_{(\text{exp})}(x) &= \frac{\frac{1}{\theta} e^{-\frac{x}{\theta}}}{e^{-\frac{x}{\theta}}} \\ &= \frac{1}{\theta}, \text{ (constant)} \end{aligned}$$

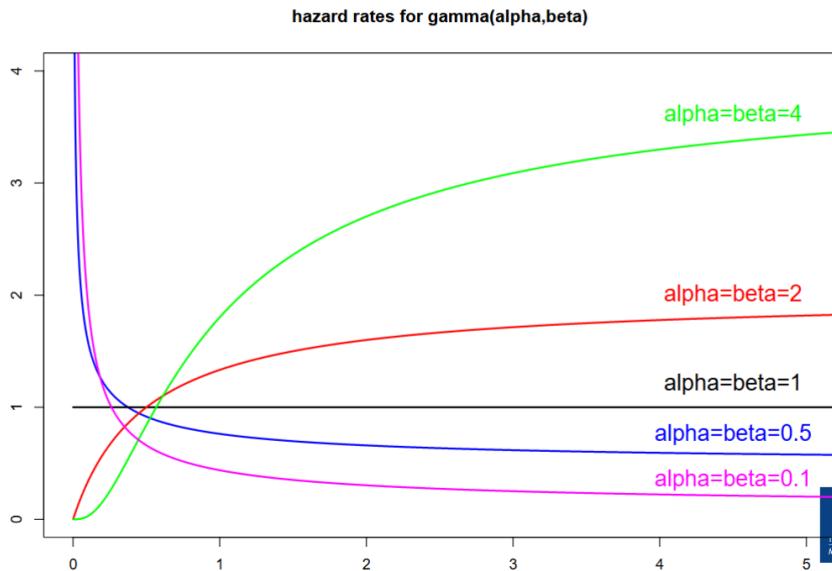
$$\begin{aligned} h_{(\text{pareto})}(x) &= \frac{\frac{\alpha \theta^\alpha}{(x+\theta)^{\alpha+1}}}{\frac{\theta^\alpha}{(x+\theta)^\alpha}} \\ &= \frac{\alpha}{x+\theta}, \text{ (decreasing in } x \text{ so heavier tail)} \end{aligned}$$

Example: Limiting density ratio - Gamma



- $\alpha > 1 \implies$ thinner tail than exponential
- $\alpha < 1 \implies$ fatter tail than exponential

Example: Hazard rate function - Gamma with fixed mean



6.2 Framework: n-block maximum

- Consider iid random variables $X_i, i = 1, \dots,$, with df F
- Denote the order statistics

$$X_{n,n} \leq X_{n-1,n} \leq \dots \leq X_{1,n}$$

such that

$$X_{n,n} = \min(X_1, \dots, X_n) \text{ and } X_{1,n} = \max(X_1, \dots, X_n)$$

6.2.1 Parallel with CLT

- In the exponential example, the normalised maximum

$$\frac{X_{1,n}}{\text{mean of } X} - \log n$$

for infinite sample n stabilised to a Gumbel distribution Λ .

- This generalises to any distribution X – the limiting distribution of the normalised maximum then becomes a Generalised Extreme Value (GEV) distribution.

6.3 GEV Distribution

$$H_{\gamma; \mu, \sigma}(x) = \begin{cases} e^{-(1+\gamma \frac{x-\mu}{\sigma})^{-\frac{1}{\gamma}}} & , \gamma \neq 0 \\ e^{-e^{-\frac{x-\mu}{\sigma}}} & , \gamma = 0 \end{cases}$$

$\mu \in \mathbb{R}$ and $\sigma > 0$

There are three cases:

- $\gamma < 0$: upper bounded Weibull with $x < \mu - \sigma/\gamma$
- $\gamma = 0$: Gumbel with $x \in \mathbb{R}$
- $\gamma > 0$: Fréchet with $x > \mu - \sigma/\gamma$ (heavy tailed distributions, right tail)

Table 3: GEV distributions (for the maximum value) corresponding to common loss distributions

Type	WEIBULL $\gamma < 0$	GUMBEL $\gamma = 0$	FRÉCHET $\gamma > 0$
Underlying Distribution	Beta Uniform Triangular	Chi-square Exponential Gamma Log-normal Normal Weibull	Burr F Log-gamma Pareto t
Range of values permitted	$x \leq \mu - \frac{\sigma}{\gamma}$	$-\infty < x < \infty$	$x \geq \mu - \frac{\sigma}{\gamma}$

6.3.1 Example: Exponential

For underlying cdf $F(x) = 1 - e^{-\beta x}$, for $\beta > 0$ and $x \geq 0$, the GEV form H can be obtained by:

$$\begin{aligned} \lim_{x \rightarrow \infty} \Pr \left[\frac{X_{1,n} - b_n}{a_n} \leq x \right] &= \lim_{x \rightarrow \infty} \Pr[X_{1,n} \leq a_n x + b_n] \\ &= \lim_{x \rightarrow \infty} F(a_n x + b_n)^n \\ &= \lim_{x \rightarrow \infty} \left(1 - e^{-\beta(a_n x + b_n)} \right)^n \\ &= \lim_{x \rightarrow \infty} \left(1 - \frac{e^{-\beta x}}{n} \right)^n = e^{-e^{-\beta x}} \end{aligned}$$

by selecting normalising constants $a_n = \frac{1}{\beta}$, $b_n = \frac{\log n}{\beta}$ and setting $\mu = 0$, $\sigma = 1$, where the range of x is given by $x \geq \lim_{x \rightarrow \infty} -\frac{b_n}{a_n} = -\infty$, i.e. $x \in \mathbb{R}$.

The limiting distribution of the normalised maximum from an underlying Exponential distribution is Gumbel from the GEV family.

6.3.2 Example: Pareto

For underlying cdf $F(x) = 1 - \left(\frac{\theta}{\theta+x}\right)^\alpha$ for $\alpha > 0$, $\theta > 0$ and $x \geq 0$, the GEV form H can be obtained by:

$$\begin{aligned} \lim_{x \rightarrow \infty} \Pr \left[\frac{X_{1,n} - b_n}{a_n} \leq x \right] &= \lim_{x \rightarrow \infty} \Pr[X_{1,n} \leq a_n x + b_n] \\ &= \lim_{x \rightarrow \infty} F(a_n x + b_n)^n \\ &= \lim_{x \rightarrow \infty} \left(1 - \left(\frac{\theta}{\theta + (a_n x + b_n)} \right)^\alpha \right)^n \\ &= \lim_{x \rightarrow \infty} \left(1 - \frac{1}{n} \left(1 + \frac{x}{\alpha} \right)^{-\alpha} \right)^n = e^{-(1+\frac{x}{\alpha})^{-\alpha}} \end{aligned}$$

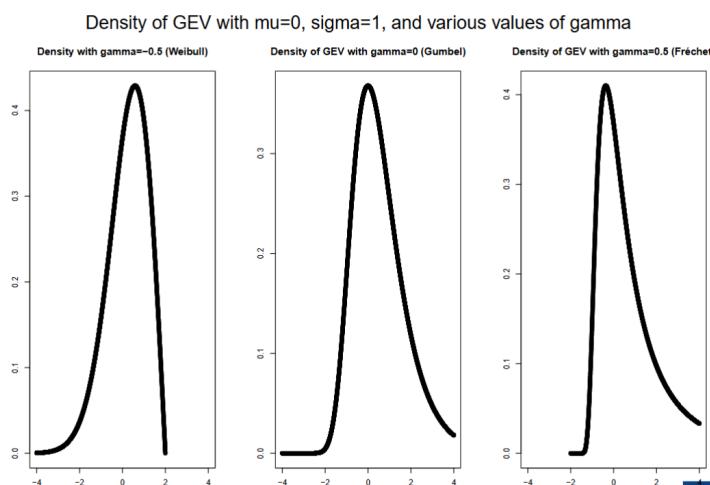
by selecting normalising constants $a_n = \frac{\theta n^{\frac{1}{\alpha}}}{\alpha}$, $b_n = \theta n^{\frac{1}{\alpha}} - \theta$ and setting $\mu = 0$ and $\sigma = 1$, where the range of x is given by $x \geq \lim_{x \rightarrow \infty} -\frac{b_n}{a_n} = -\alpha$, i.e. $x \in [-\alpha, \infty)$.

The limiting distribution of the normalised maximum from an underlying Pareto distribution is Fréchet from the GEV family.

6.3.3 GEV Density Plot of Gamma (in R)

```
GenEV <- function(x, alpha, beta, gamma) {
  1/beta * (1 + gamma * (x - alpha)/beta)^(-(1 + 1/gamma)) * exp(-((1 +
  gamma * (x - alpha)/beta)^(-1/gamma)))
}

par(mfrow = c(1, 3), oma = c(0, 0, 3, 0))
plot(-4000:2000/1000, GenEV(-4000:2000/1000, 0, 1, -0.5),
main = "Density with gamma=-0.5 (Weibull)", xlim = c(-4,
4), xlab = "", ylab = "", cex = 1) ## gamma = -0.5 (Weibull)
plot(-4000:4000/1000, GenEV(-4000:4000/1000, 0, 1, 1e-05),
main = "Density of GEV with gamma=0 (Gumbel)", xlim = c(-4,
4), xlab = "", ylab = "", cex = 1) ## gamma = 0 (Gumbel)
plot(-2000:4000/1000, GenEV(-2000:4000/1000, 0, 1, 0.5),
main = "Density of GEV with gamma=0.5 (Fréchet)", xlim = c(-4,
4), xlab = "", ylab = "", cex = 1) ## gamma = 0.5 (Frechet)
mtext("Density of GEV with mu=0, sigma=1, and various values of gamma",
outer = TRUE, cex = 1.5)
```



6.3.4 Identifying a Fréchet GEV ($\gamma > 0$)

- A function $L(x)$ is slowly varying at ∞ if

$$\lim_{x \rightarrow \infty} \frac{L(tx)}{L(x)} = 1, \quad t > 0$$

- A function $h(x)$ is regularly varying at ∞ with index ρ if

$$\lim_{x \rightarrow \infty} \frac{h(tx)}{h(x)} = t^\rho \Leftrightarrow h(x) = x^\rho L(x), \quad t > 0$$

The limiting distribution of the normalised maximum from an underlying df F is of Fréchet type ($\gamma > 0$) if its survival function:

$$\tilde{F}(x) = 1 - F(x) = x^{-\frac{1}{\gamma}} L(x)$$

The rate of decay $\frac{1}{\gamma}$ is often known as the tail index of the distribution. We also know $\mathbb{E}[X^k] = \infty$ for $k > \frac{1}{\gamma}$.

Example: Pareto distribution

$$\begin{aligned} F(x) &= 1 - \left(\frac{\theta}{\theta + x} \right)^\alpha, \quad \alpha > 0, \theta > 0, x \geq 0 \\ \tilde{F}(x) &= \left(\frac{\theta}{\theta + x} \right)^\alpha \\ &= x^{-\alpha} \left(\frac{\theta}{\frac{\theta}{x} + 1} \right)^\alpha \\ &= x^{-\alpha} \left(\frac{\theta x}{\theta + x} \right)^\alpha = x^{-\alpha} L(x) \end{aligned}$$

for $L(x) = \left(\frac{\theta x}{\theta + x} \right)^\alpha$, the rate of decay $\frac{1}{\gamma} = \alpha$, so the limiting distribution is of Fréchet type with $\gamma = \frac{1}{\alpha}$.

6.3.5 Identifying a Gumbel GEV ($\gamma = 0$)

The limiting distribution of the normalised maximum from an underlying df F is of Gumbel type ($\gamma = 0$) if it has finite moments of any positive order, i.e. $\mathbb{E}[X] < \infty$ for $k > 0$. Examples include Exponential, Normal, Gamma, Weibull, and Lognormal.

6.3.6 Identifying a upper-bounded Weibull ($\gamma < 0$)

The limiting distribution of the normalised maximum from an underlying df F is of upper-bounded Weibull type ($\gamma < 0$) if it has finite right end point (i.e. $x_F < \infty$) where the right endpoint is defined as $x_F = \sup\{x \in \mathbb{R} : F(x) < 1\}$.

6.3.7 Estimation: Block Maxima (in R)

- The EVD is a distribution of maxima
- To fit we need a sample of maxima
- To achieve this, the data is transformed into **block maxima**
 1. assume we have n data points (say, monthly data over 1,000 years, so $n=12,000$)
 2. consider blocks of length m (say, all observations in a given year such that $m=12$)
 3. the block maxima are then equal to the maximum observation within a block (the maximum value within each of the 1,000 years)

4. we have then a sample size of n/m (*our sample size of maxima is of size $12000/12=1,000$*)
- The block maxima are then fitted to the EVD
 - The larger the block size, the closer to the asymptotic distribution we get, but the less data we have (conflicting requirements)
 - In R we use `fevd()` from package `extRemes`

Example: with Data (in R):

Consider three datasets with $n = 12,000$, given in months, with columns named `beta` (simulated from beta), `gamm` (simulated from gamma), `logg` (simulated from log-gamma).

Creating block maxima

```
claims <- as_tibble(read_excel("simulated-claims.xlsx"))
# block maxima index
claims$block <- (claims$month - 1)%/12 + 1
# %/ gives the integer part of the result of the division
blockmax <- tibble(betablock = aggregate(beta ~ block, claims,
max)$beta, gammablock = aggregate(gamm ~ block, claims, max)$gamm,
loggblock = aggregate(logg ~ block, claims, max)$logg)

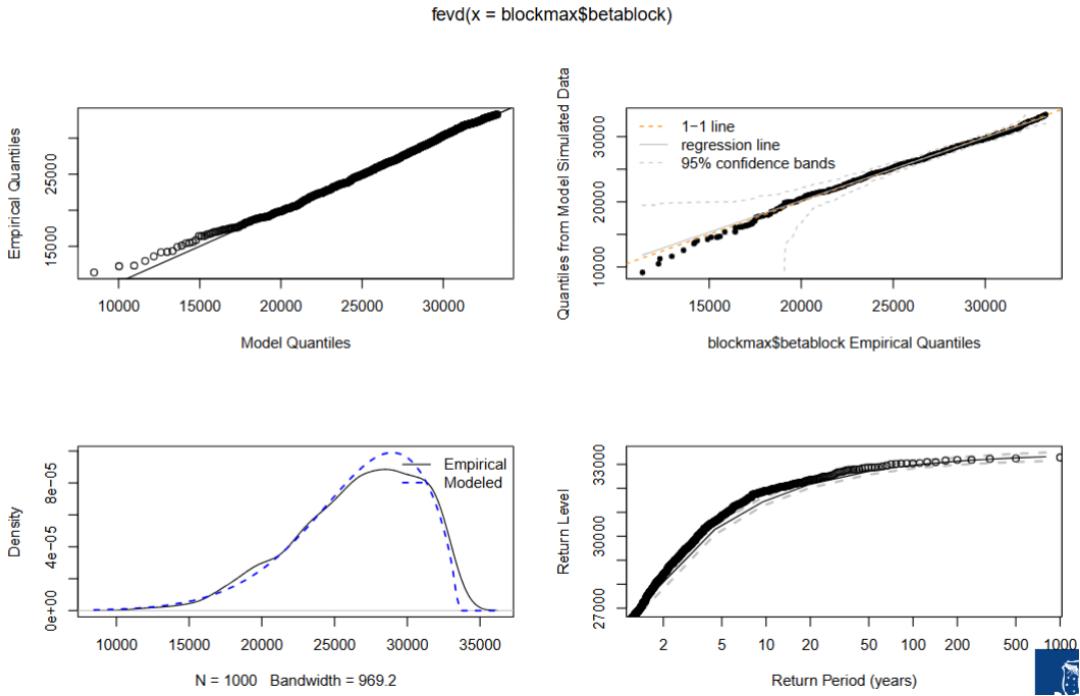
# Can also plot the same for the gamma and log-gamma blocks
par(mfrow = c(1, 2))
plot(density(claims$beta), main = "Density of the beta claims",
xlab = "Claim amounts ($)", xlim = c(0, max(claims$beta)))
plot(density(blockmax$betablock),
main = "Density of the beta block maxima",
xlab = "Maximums over consecutive periods of 12 months ($)",
xlim = c(0, max(claims$beta)))

library(extRemes)
fit.beta <- fevd(blockmax$betablock)
fit.beta # shows parameter estimates, gamma < 0
plot(fit.beta) # plots goodness of fit analysis

fit.gamm <- fevd(blockmax$gammablock)
fit.gamm # shows parameter estimates, gamma = 0 can be considered due to s.e.
plot(fit.gamm) # plots goodness of fit analysis

fit.logg <- fevd(blockmax$loggblock)
fit.logg # shows parameter estimates
plot(fit.logg) # plots goodness of fit analysis

# Example of plot(fit.beta)
```



- Top-left: if good fit dots should almost be a straight line,
- Top-right: black dots should fit to the fitted line,
- Bottom-left: depends on what the GEV looks like (beta gives upper-bounded weibull dist. and it is bounded around 33,000),
- Bottom-right: dots should closely follow line.

6.4 GPD Distribution

- The tail is the distribution beyond a point. We characterise the tail with the df of the excess over threshold u (**threshold exceedance**)

$$F_u(x) = \Pr[Y - u \leq x | Y > u] = \frac{F(x + u) - F(u)}{1 - F(u)}, \quad 0 \leq x < x_F - u$$

- For high u , $F_u(x)$, can be *approximated* by a Generalised Pareto Distribution (GPD)

$$G_{\gamma, \sigma}(x) = \begin{cases} 1 - \left(1 + \gamma \frac{x}{\sigma}\right)^{-\frac{1}{\gamma}} & \gamma \neq 0 \\ 1 - e^{-\frac{x}{\sigma}} & \gamma = 0 \end{cases}$$

for a scale parameter $\sigma > 0$.

We distinguish again three cases:

- $\gamma < 0$ (upper bounded, also referred to as "Pareto Type II"): **light tail** $x \in (0, \sigma/|\gamma|)$
- $\gamma = 0$ (exponential): base case $x \in (0, \infty)$
- $\gamma > 0$ (Pareto): **heavy tail** $x \in (0, \infty)$

Moments

The first central moments of X (in excess of u) are

$$\mathbb{E}[X] = \frac{\sigma}{1 - \gamma}, \quad \gamma < 1$$

$$\text{Var}(X) = \frac{\sigma^2}{(1-\gamma)^2(1-2\gamma)}, \gamma < 1/2$$

Note:

- The first k central moments exist only for $\gamma < 1/k$.
- $\mathbb{E}[X]$ is the stop loss premium (since X is the excess over threshold u).

6.4.1 Estimation: choice of threshold u (in R)

We seek to estimate the tail $\bar{F}(u+x) = \bar{F}(u)\bar{F}_u(x)$. To do this, we need to estimate:

- Probability of exceeding u : $\widehat{\bar{F}(u)} = \frac{n_u}{n}$, which will be more accurate for u not too large.
- Then for a given value of u : $\widehat{\bar{F}_u(x)} = \bar{G}_{\hat{\gamma}, \hat{\sigma}}(x)$, where $\hat{\gamma}(u)$ and $\hat{\sigma}(u)$ are estimated from the data in the tail above u (via MLE). This approximation will work well only for large u .

The choice of u presents conflicting requirements for u :

- Larger u will lead to a better approximation from a distributional perspective
- but larger u reduces the amount of data to estimate $\bar{F}(u)$, γ and σ

The choice is often also based on the following further considerations:

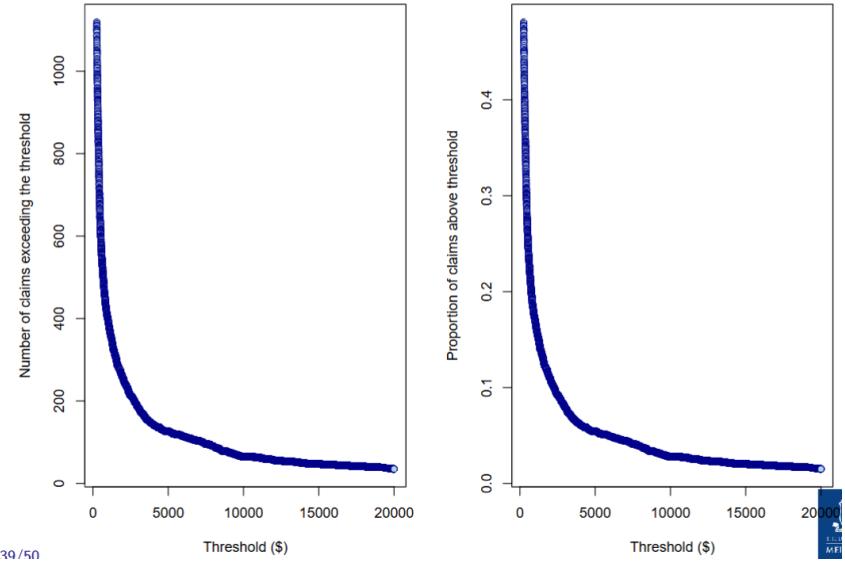
- mean-excess plot: the slope should have the same sign as γ
- log-log plot: linear for heavy tails
- stability of $\hat{\gamma}$ for different choices of u
- Hill plot: stability of Hill estimator $\hat{\alpha}$
- goodness of fit for different choices of u

Example: SUVA dataset data

Size of tail (want to choose threshold which is about 5% of values above threshold, e.g. 5000 in this example):

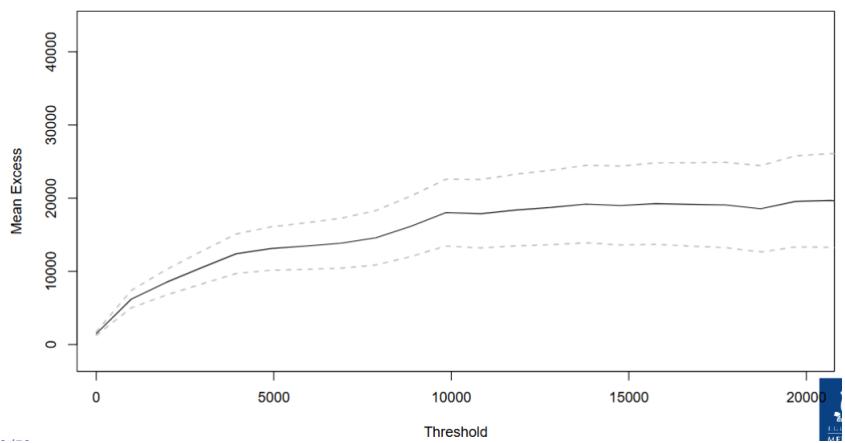
```
numbexc <- c()
for (i in 250:20000) {
  numbexc <- c(numbexc, length(SUVA$medcosts[SUVA$medcosts > i]))
}

par(mfrow=c(1,2))
plot(250:20000, numbexc, xlab = "Threshold", ylab = "Number of claims exceeding the threshold",
  col="darkblue", bg="lightblue", pch=21)
plot(250:20000, numbexc/length(SUVA$medcosts), xlab = "Threshold", ylab = "Proportion of claims
  above threshold", col = "darkblue", bg = "lightblue", pch = 21)
```



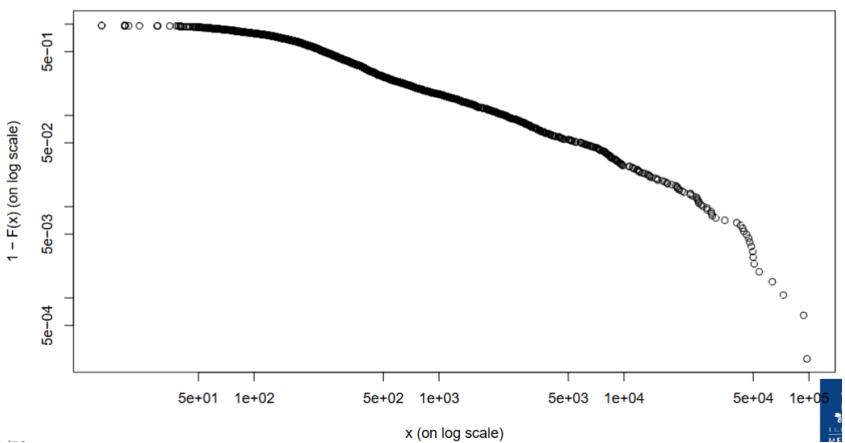
Mean-excess plot (if upwards sloping means heavy tail so $\gamma > 0$, if downwards sloping means lighter tail so $\gamma < 0$):

```
extRemes::mrlplot(SUVA$medcosts, xlim = c(250, 20000))
```



Empirical \bar{F} on log-log scale:

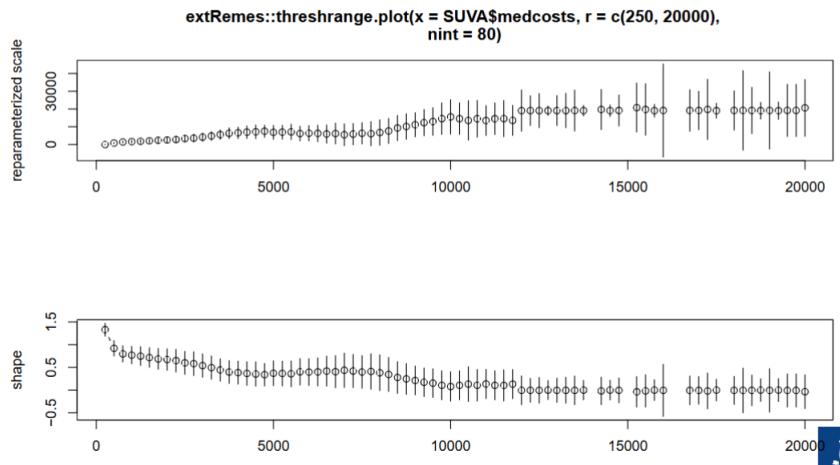
```
evir::emplot(SUVA$medcosts, alog = "xy", labels = TRUE)
```



good potential candidate for threshold is a bit before 5000, before line goes funny

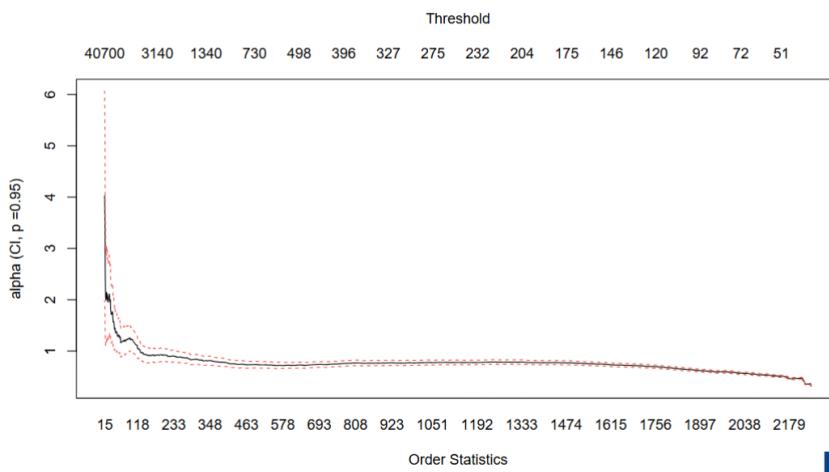
Stability of $\hat{\gamma}$, only bottom graph is important (top concerns the scale σ so ignore), need to find an area where the plot is flat then that is a good threshold value and the y-axis indicates the γ :

```
extRemes::threshrange.plot(SUVA$medcosts, r = c(250, 20000), nint = 80)
```



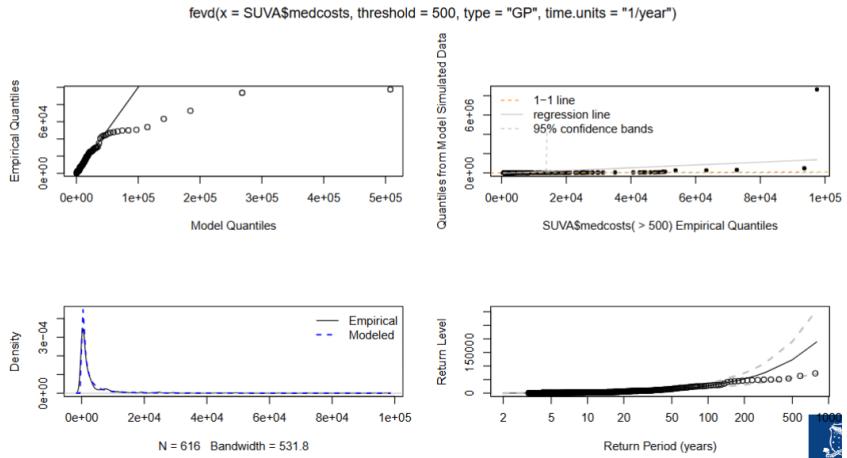
Hill plot, Stability of $\hat{\gamma}$ is more important than this, if need to just look at where it is flat just before it shoots up on the left:

```
evir::hill(SUVA$medcosts) # note that alpha is inverse of gamma
```



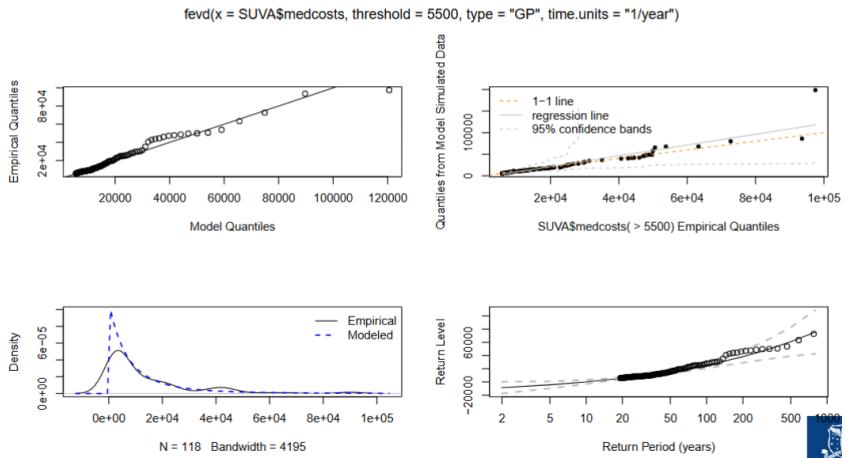
Estimation results for $u = 500$

```
fit.SUVA.500 <- fevd(SUVA$medcosts, threshold = 500, type = "GP",
time.units = "1/year")
fit.SUVA.500 # shows estimated parameters of shape and scale
numbexc[501 - 250] # n_u (noting that numbexc[1] = n_250)
## 616
plot(fit.SUVA.500) # top 2 graphs show its bad choice
```



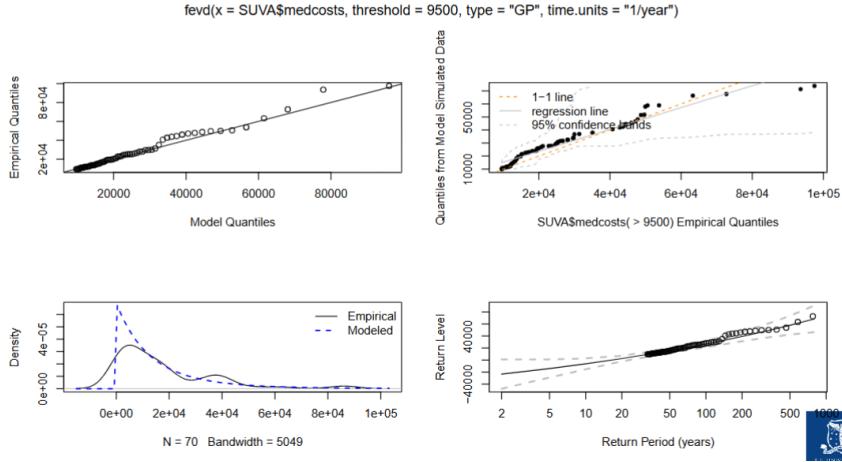
Estimation results for $u = 5500$

```
fit.SUVA.5500 <- fevd(SUVA$medcosts, threshold = 5500, type = "GP",
time.units = "1/year")
fit.SUVA.5500 # shows estimated parameters of shape and scale
numbexc[5501 - 250] # n_u
## 118
plot(fit.SUVA.5500) # top 2 graphs show its bad choice
```



Estimation results for $u = 9500$

```
fit.SUVA.9500 <- fevd(SUVA$medcosts, threshold = 9500, type = "GP",
time.units = "1/year")
fit.SUVA.9500 # shows estimated parameters of shape and scale
numbexc[9501 - 250] # n_u
## 70
plot(fit.SUVA.9500) # top 2 graphs show its bad choice
```



Return level in R:

```
library(extRemes)

# Example data: say "data" is your numeric vector of observations
fit <- fevd(data, threshold = u, type = "GP")
return.level(fit, return.period = T)
# T = return period, e.g. 100 for 100 year return
```

The return level is the value you expect to occur once in every T periods (where T is the return period, e.g. $T = 100$ years).

QQ Plot of fit (just QQ by itself):

```
# QQ Plot
plot(fit, type="qq", main="GPD QQ-plot: empirical vs theoretical")
abline(0,1, lty="dashed", col="blue")
```

7 Module 7 – Time Series Fundamentals

7.1 Preliminaries and White Noise Basics

A time series is a sequence of rv's x_1, x_2, x_3, \dots , denoted $\{x_t\}$. **White Noise**: w_t is uncorrelated (over t) with mean 0 and finite variance σ_w^2 , it can be *iid* and Gaussian, or just *iid*.

Plot a gaussian white noise series with a 3-point moving average ($v_t = \frac{1}{3}(w_{t-1} + w_t + w_{t+1})$):

```
w = rnorm(500, 0, 1) # 500 N(0,1) variates
plot.ts(w, ylim=c(-3,3), main="white noise")
v = stats::filter(w, sides=2, filter=rep(1/3,3)) # moving average
plot.ts(v, ylim=c(-3,3), main="moving average")
```

Plot an auto-regression with gaussian white noise $x_t = x_{t-1} - 0.9x_{t-2} + w_t$ for $t = 1, \dots, 500$. We also need 2 initial conditions since the time series goes 2 periods back (to avoid this problem we run the AR for longer than needed to avoid this issue and remove first 50 observations).

```
w = rnorm(550, 0, 1)
x = stats::filter(w, filter = c(1, -0.9), method="recursive")[-(1:50)] # remove first 50
plot.ts(x, ylab="autoregression", main=expression(x[t] == x[t-1] - 0.9 * x[t-2] + w[t]))
```

Notes of R: `filter(x, filter, method = c("convolution"), sides=2)`

- `filter`: vector of weights (in reverse time order),
- `method`: default is "convolution" (for MA) and "recursive" (for AR),
- `sides`: 1 for past values only, and 2 if weights are centred around lag 0.

Random walk with drift

The random walk with drift looks back only one time unit:

$$x_t = \delta + x_{t-1} + w_t = \delta t + \sum_{j=1}^t w_j, \text{ initial condition } x_0 = 0$$

- If $\delta = 0$ then it is a random walk
- $\Delta x_t = x_{t+1} - x_t = w_{t+1}$ so only depends on random noise and nothing prior
- $\mu_{x_t} = \mathbb{E}[x_t] = \delta t$ and $\text{Var}(x_t) = t\sigma_w^2$, so mean and variance varies with time
- $\gamma_x(s, t) = \text{Cov}(\sum_{j=1}^s w_j, \sum_{k=1}^t w_k) = \min\{s, t\}\sigma_w^2$

7.1.1 Auto-covariance and correlation functions

Auto-covariance function:

$$\gamma_x(s, t) = \text{Cov}(x_s, x_t) = \mathbb{E}[(x_s - \mu_{x_s})(x_t - \mu_{x_t})]$$

A smooth series \rightarrow large γ even for large lag, for choppy series \rightarrow γ is nearly 0 for large lags.

Cross-covariance function:

$$\gamma_{xy}(s, t) = \text{Cov}(x_s, y_t) = \mathbb{E}[(x_s - \mu_{x_s})(y_t - \mu_{y_t})]$$

Useful Covariance Property:

If $U = \sum_{j=1}^m a_j X_j$ and $V = \sum_{k=1}^r b_k Y_k$, then:

$$\text{Cov}(U, V) = \sum_{j=1}^m \sum_{k=1}^r a_j b_k \text{Cov}(X_j, Y_k)$$

Auto-correlation function:

$$-1 \leq \rho(s, t) = \frac{\gamma(s, t)}{\sqrt{\gamma(s, s)\gamma(t, t)}} \leq 1$$

Measurers the *linear* predictability of the series at time t , say x_t , using only the value of x_s .

Cross-correlation function:

$$-1 \leq \rho_{xy}(s, t) = \frac{\gamma_{xy}(s, t)}{\sqrt{\gamma_x(s, s)\gamma_y(t, t)}} \leq 1$$

7.2 Stationarity

7.2.1 Strictly Stationary

A time series which the probabilistic behaviour of every collection of values $\{x_{t_1}, \dots, x_{t_k}\}$ is identical to that of the time shifted (for any h) $\{x_{t_1+h}, \dots, x_{t_k+h}\}$. That is:

$$\mathbb{P}(x_{t_1} \leq c_1, \dots, x_{t_k} \leq c_k) = \mathbb{P}(x_{t_1+h} \leq c_1, \dots, x_{t_k+h} \leq c_k)$$

For all $k = 1, 2, \dots$, all the time points t_i , numbers c_i and **all time shifts** $h = 0, \pm 1, \pm 2, \dots$ This implies:

- Identical marginals of dimensions $< k$ for any shift h
- Constant mean: $\mu_{xs} = \mu_{xt} \equiv \mu$
- The auto-covariance function depends only on lag $t - s$: $\gamma(s, t) = \gamma(s + h, t + h)$
- A weakly stationary Gaussian time series is also strictly stationary.

7.2.2 Weak Stationary

Time series is weakly stationary when it is a finite variance process such that:

1. The mean value function, μ_{xt} is **constant** and does not depend on time t , and
2. The auto-covariance function, $\gamma(s, t)$ depends on s and t **only through their difference** $|s - t|$.

7.2.3 Properties

- By condition 1 (constant mean): $\mu_t = \mu$
- By condition 2 (covariance only depends on lag): $\gamma(t+h, t) = \text{Cov}(\mathbf{x}_{t+h}, \mathbf{x}_t) = \text{Cov}(x_h, x_0) = \gamma(h, 0) \equiv \gamma(\mathbf{h}) = \mathbb{E}[(x_{t+h} - \mu)(x_t - \mu)]$
- $\gamma(\mathbf{h})$ is non-negative definite: $0 \leq \text{Var}(a_1 x_1 + \dots + a_n x_n) = \sum_{j=1}^n \sum_{j=1}^n a_j a_k \gamma(j - k)$
- Also, $|\gamma(\mathbf{h})| \leq \gamma(0)$ and $\gamma(\mathbf{h}) = \gamma(-\mathbf{h})$
- The ACF of the stationary time series becomes: $-1 \leq \rho(\mathbf{h}) = \frac{\gamma(t+h, t)}{\sqrt{\gamma(t+h, t+h)\gamma(t, t)}} = \frac{\gamma(\mathbf{h})}{\gamma(0)} \leq 1$.

Examples of (non)-stationarity

Stationary white noise:

We have $\mu_{w_t} = 0$ and $\gamma_w(h) = \text{Cov}(w_{t+h}, w_t) = \begin{cases} \sigma_w^2 & h = 0, \\ 0 & h \neq 0 \end{cases}$,

These are both independent of time. Hence, the white noise satisfies both conditions and is (weakly) stationary. Furthermore, $\rho_w(h) = \begin{cases} 1 & h = 0, \\ 0 & h \neq 0. \end{cases}$

If in addition $w_t \sim \text{iidN}(0, \sigma_w^2)$, then it is also strictly stationary.

Stationary moving average: for 3-point MA we have

$$\mu_{vt} = 0 \text{ and } \gamma_v(h) = \begin{cases} \frac{3}{9}\sigma_w^2 & h = 0, \\ \frac{2}{9}\sigma_w^2 & h = \pm 1, \\ \frac{1}{9}\sigma_w^2 & h = \pm 2, \\ 0 & |h| > 2 \end{cases}$$

Which are both independent of time. Hence, the 3-point MA satisfies both conditions and is stationary. Furthermore the ACF is symmetric around lag 0.

Non-stationary random walk: for the random walk model $x_t = \delta t + \sum_{j=1}^t w_j$ we have,

$$\mu_{xt} = \delta t \text{ and } \gamma(s, t) = \min\{s, t\}\sigma_w^2$$

The mean depends on time t and the auto covariance function depends on s and t (not just their difference), so the random walk is NOT stationary.

Furthermore, $\text{Var}(x_t) = \gamma_x(t, t) = t\sigma_w^2$ which increases without bound as $t \rightarrow \infty$.

7.2.4 Trend Stationarity

- If the second condition (on the ACF that it only depends on lag) is satisfied, but not the first condition (on the mean being constant and independent of time), we have Trend Stationarity.
- The model has a stationary behaviour around its trend.

Example: if $x_t = \alpha + \beta t + y_t$ (assuming y_t is stationary):

$$\mu_{xt} = \mathbb{E}[x_t] = \alpha + \beta t + \mu_y \text{ and } \gamma_x(h) = \gamma_y(h)$$

Therefore the mean is *not* independent of time and the auto-covariance function is independent of time.

7.2.5 Joint Stationarity

Two time series x_t and y_t are jointly stationary if they are each stationary AND the cross covariance function is a function of only lag h .

$$\gamma_{xy}(h) = \text{Cov}(x_{t+h}, y_t) = \mathbb{E}[(x_{t+h} - \mu_x)(y_t - \mu_y)] \text{ and } 1 \leq \rho_{xy}(h) = \frac{\gamma_{xy}(h)}{\sqrt{\gamma_x(0)\gamma_y(0)}} \leq 1$$

They aren't symmetric around zero. However, we have $\gamma_{xy}(h) = \gamma_{yx}(-h)$ and $\rho_{xy}(h) = \rho_{yx}(-h)$.

Example of joint stationarity, consider $x_t = w_t + w_{t-1}$ and $y_t = w_t - w_{t-1}$, we know then:

$$\begin{aligned} \gamma_x(0) &= \gamma_y(0) = 2\sigma_w^2 \\ \gamma_x(-1) &= \gamma_x(1) = \sigma_w^2 \\ \gamma_y(-1) &= \gamma_y(1) = -\sigma_w^2 \end{aligned}$$

and

$$\gamma_{xy}(-1) = -\sigma_w^2, \quad \gamma_{xy}(0) = 0, \quad \text{and } \gamma_{xy}(1) = \sigma_w^2$$

so that:

$$\rho_{xy}(h) = \begin{cases} 0 & h = 0, \\ 1/2 & h = 1, \\ -1/2 & h = -1, \\ 0 & |h| \geq 2 \end{cases}$$

Which depends only on the lag h , so both series are jointly stationary.

7.2.6 Linear processes

A *linear process*, x_t , is defined to be a linear combination of white noise variates, and is given by:

$$x_t = \mu + \sum_{j=-\infty}^{\infty} \Psi_j w_{t-j}, \quad \sum_{j=-\infty}^{\infty} |\Psi_j| < \infty$$

Properties:

- The auto-covariance is given by: $\gamma_x(h) = \sigma_w^2 \sum_{j=-\infty}^{\infty} \Psi_{j+h} \Psi_j$ for $h \geq 0$
- It has finite variance if $\sum_{j=-\infty}^{\infty} \Psi_j^2 < \infty$
- x_t depends on the future (if $j < 0$), the present (if $j = 0$), and the past (if $j > 0$)
- If a process does NOT depend on the future it is **casual**: $\Psi_j = 0$ for $j < 0$

7.3 Estimation of Correlation

7.3.1 Sample mean

Assuming that a time series is stationary the mean function $\mu_t = \mu$ is constant so that we can estimate it by the sample mean:

$$\bar{x} = \frac{1}{n} \sum_{t=1}^n x_t$$

This estimator is unbiased, $\mathbb{E}[\bar{x}] = \mu$ and has the standard error of: $\text{Var}(\bar{x}) = \frac{1}{n^2} \text{Cov}(\sum_{t=1}^n x_t, \sum_{s=1}^n x_s) = \frac{1}{n} \sum_{h=-n}^n (1 - \frac{|h|}{n}) \gamma_x(h)$

7.3.2 Sample autocovariance function

$$\hat{\gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(x_t - \bar{x}) \text{ with } \hat{\gamma}(-h) = \hat{\gamma}(h) \text{ for } h = 0, 1, \dots, n-1$$

- The estimator is unbiased
- The sum is bounded to $n - h$ because x_{t+h} is not available for $t + h > n$

7.3.3 Sample auto-correlation function

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}$$

Under certain conditions, if x_t is white noise, then for large n , the SACF is approximately normally distributed with $\mu = 0$ and $\sigma_{\hat{\rho}(h)} = \frac{1}{\sqrt{n}}$

Testing for significance of auto-correlation

We can test whether lagged observations are uncorrelated (which is a requirement for white noise):

- Test for significance of the $\hat{\rho}$'s at different lags: check how many $\hat{\rho}$ values lie outside of the interval $\pm 2/\sqrt{n}$ (a 95% C.I.)
- Should expect approximately **1 out of 20** to lie outside the interval if the sequence is white noise, many more would invalidate the whiteness assumption
- The R function `acf` automatically displays those bounds with dashed blue lines

Example with SOI data

```
acf(soi, main="Sample Auto-correlation function of SOI")
r <- round(acf(soi, 6, plot=FALSE)$acf[-1], 3) # first 6 sample acf
# The series is clearly NOT white noise as lots of lines beyond dashed line
```

```
# Visualising dependence of SOI pairs with 6 month lag
plot(stats::lag(soi, -6), soi, main="SOI pairs of values 6 months apart")
legend("topleft", legend = r[6])
```

7.3.4 Sample cross-covariances and cross-correlations

Sample cross-covariance function

$$\hat{\gamma}_{xy}(h) = \frac{1}{n} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(y_t - \bar{y})$$

where $\hat{\gamma}_{xy}(-h) = \hat{\gamma}_{xy}(h)$ determines the function for negative lags

Sample cross-correlation function

$$-1 \leq \hat{\rho}_{xy}(h) = \frac{\hat{\gamma}_{xy}(h)}{\sqrt{\hat{\gamma}_x(0)\hat{\gamma}_y(0)}} \leq 1$$

Note: graphical examinations of this function provide information about the leading or lagging relations in the data as shown below.

7.3.5 Prediction using cross-correlation

A lagging relation between two series x_t and y_t may be exploited for predictions. For instance if

$$y_t = Ax_{t-\ell} + w_t$$

x_t is said to **lead** y_t for $\ell > 0$, and is said to **lag** y_t for $\ell < 0$.

If the relation above holds true, then the lag ℓ can be inferred from the cross-covariance of y_{t+h} on x_t denoted $\gamma_{yx}(h)$:

- If w_t is uncorrelated with x_t then, $\gamma_{yx}(h) = \text{Cov}(y_{t+h}, x_t) = A\gamma_x(h - \ell)$
- We know $|\gamma_x(h - \ell)| \leq \gamma_x(0)$, and so $\ell = h$ is the peak of $\gamma_{yx}(h)$. Also $h = \ell$ will be positive if x_t leads y_t or negative if x_t lags y_t
- Done using `ccf` in R

SOI and Recruitment correlation analysis

```
ccf(rec, soi, 48, main = "SOI vs recruitment", ylab = "CCF") # so y = rec and x = soi
# The plot has a peak at h=6, suggesting SOI leads recruitment by 6 months
```

8 Module 8 - Time Series Smoothing and Exploratory Analysis

8.1 Regression in time series context

Assume that some dependent time series x_t , can be make up of a collection of q independent time series $z_{t1}, z_{t2}, \dots, z_{tq}$ where we first regard the inputs as fixed and known. Therefore we have:

$$x_t = \beta_0 + \beta_1 z_{t1} + \beta_2 z_{t2} + \dots + \beta_q z_{tq} + w_t$$

Example regression with time variable

The monthly price x_t of a chicken in the US, we model the *trend* with a linear regression: $x_t = \beta_0 + \beta_1 t + w_t$, where z_t are months in the data ($q = 1$), we also assume that the w_t are iid gaussian white noise (which may not be true in reality).

```
summary(fit <- lm(chicken~time(chicken), na.action=NULL))
# Estimate Std. Error t.value
# (Intercept) -7131.02 162.41 -43.9
# time(chicken) 3.59 0.08 44.4
# --
# Residual standard error: 4.7 on 178 degrees of freedom

# Note that beta hat 1 = 3.59 meaning an increase of about 3.6 cents per year with
# a standard error of 0.081

plot(chicken, ylab="cents per pound")
abline(fit) # add the fitted line
```

Example regression with lagged variables

Previously we found that SOI leads Recruitment by six months. The relationship is not exactly linear by we consider $R_t = \beta_0 + \beta_1 S_{t-6} + w_t$. Under fitting from the following R code we get $\hat{R}_t = 65.790 - 44.283(2.781)S_{t-6}$

```
fish = ts.intersect(rec, soiL6=lag(soi,-6), dframe=TRUE) # to match up lagged timings
summary(fit1 <- lm(rec~soiL6, data=fish, na.action=NULL))
# Estimate Std. Error t.value Pr(>|t|)
# (Intercept) 65.790 1.088 60.47 <2e-16 ***
# time(chicken) -44.283 2.781 -15.92 <2e-16 ***
# The low p-values imply that SOI is a strong predictor of recruitment

# The same output without ts.intersect can be done via:
library(dynlm)
summary(fit2 <- dynlm(rec ~ L(soi, 6)))
```

8.1.1 De-trending via regression and differencing

Context

$x_t = \mu_t + y_t$ is called *trend stationary* if y_t is stationary.

"Removing" μ_t is called **de-trending**, and there are two types:

1. via regression: fitting a regression model for $\hat{\mu}_t$ (parametric)
2. via differencing: modifying the series by looking at differences over time, rather than absolute values (non-parametric)
 - Good if μ_t does not look like a linear trend
 - $\nabla x_t \equiv x_t - x_{t-1}$
 - Differencing n times (∇^n) removes a polynomial trend of degree n
3. Note: Differencing is always at least as good as regression when it comes to de-trending

Example: Chicken prices de-trended via regression

```
fit <- lm(chicken ~ time(chicken), na.action=NULL)
plot(resid(fit), type="l", main = "Price of chicken de-trended via regression")
# lots outside CI bands so not reasonably white noise
```

Note that the residuals are $x_t - \hat{\mu}_t = y_t$ which we would like to be stationary.

Example: JJ prices de-trending via regression and differencing

Consider the Johnson & Johnson data in the `astsa` dataset (use `jj`):

1. Fit a linear trend, plot it along with the original series, and examine residuals (plot them, and compute their ACF). Comment on your results.
2. Difference the data, and examine the corresponding ACF. Comment on your results.
3. Fit a quadratic trend, plot it along with the original series, and examine residuals (plot them, and compute their ACF). Comment on your results.
4. Apply a log transformation and fit a linear trend to the logged series. Plot results along with the original series, and examine residuals (plot them, and compute their ACF). Comment on your results.

```
### Part 0 - Inspect the data
library(astsa)
par(mfrow = c(2, 1))
plot(jj, type = "o", ylab = "Quarterly Earnings per Share",
main = "JJ data")
acf(jj)

### Part 1 - Fit a linear trend
trend.jj = time(jj) # time
fit.jj <- lm(jj ~ trend.jj, na.action = NULL)
summary(fit.jj)
plot(jj, main = "JJ with linear trend")
abline(fit.jj, col = "blue", lwd = 2)

par(mfrow = c(2, 1))
plot(jj - fitted(fit.jj),
main = "JJ residuals after linear trend")
acf(jj - fitted(fit.jj))
# The residual plot has a U-shape showing linear trend not good

### Part 2 - Differencing
par(mfrow = c(2, 1))
plot(diff(jj), type = "o", ylab = "Quarterly Earnings per Share",
main = "JJ data after differencing")
acf(diff(jj))
# The diff plot has increasing magnitude violating stationarity. Series seems to be
# centered around 0, so maybe log transformation useful to equalise the variance.

### Part 3 - Fit a quadratic trend
trend.jj.2 = time(jj)^2 # time
fit.jj.2 <- lm(jj ~ trend.jj + trend.jj.2, na.action = NULL)
summary(fit.jj.2)
plot(jj, main = "JJ with quadratic trend")
points(fitted(fit.jj.2), col = "blue", lwd = 1)

par(mfrow = c(2, 1))
plot(jj - fitted(fit.jj.2),
main = "JJ residuals after quadratic trend")
```

```

acf(jj - fitted(fit.jj.2))
# Quadratic trend looks okay, but still issue of increasing magnitude in residuals plot.
# Log transform should fix.

### Part 4 - Fit linear trend with log transformed data
fit.jj.log <- lm(log(jj) ~ trend.jj, na.action = NULL)
summary(fit.jj.log)
plot(jj, main = "JJ with log transformation")
points(exp(fitted(fit.jj.log)), col = "blue", lwd = 1) ## need exp()

par(mfrow = c(2, 1))
jj.log.residuals <- jj - exp(fitted(fit.jj.log))
plot(jj.log.residuals,
main = "JJ residuals after log transformation")
acf(jj.log.residuals)
# Log transform very similar to quadratic fit which makes sense, residuals start better
# than quadratic fit but still increasing magnitude (variance) and log transformation
# is too strong as residuals even become negative. None of the above
# transformations are appropriate.

### The best one is differencing of log transformed data.

```

Example: Global temperature de-trended via differencing

```

plot(gtemp_land, type="o") # Clear upward somewhat linear trend
plot(diff(gtemp_land), type="l", main="Global temps de-trended via differencing")
acf(diff(gtemp_land), 48) # only 2 in 48 outside of CI bands so reasonably white noise

```

8.2 Scatter plot matrices to visualise relationships

Transformations:

If obvious non-linear behaviour, can lead to non-stationarity, and transformations may be useful to *equalise* the variability like $y_t = \log(x_t)$.

Scatter plot matrices:

- A preliminary data processing technique, to visualise the relations between series at different lags.
- Scatter plots (one for each lag) are informative to diagnose **non-linear** relationships (unlike ACF which focuses on linear predictability only).
- They give a visual sense of which lag will lead to the best predictability
- The red lines are *locally weighted scatter plot smoothing* (lowess) lines – they help identify non-linear relationships.

Example: SOI and SOI vs Recruitment

```

lag1.plot(soi, 12) # red lines are more or less linear for lagged SOI so sample ACF is meaningful
lag2.plot(soi, rec, 8) # red lines are highly nonlinear around lags 5-8

```

Using dummy variables we can model this (where $D_t = 0$ if $S_t < 0$ (1 otherwise)).

$$\begin{aligned}
R_t &= \beta_0 + \beta_1 S_{t-6} + \beta_2 D_{t-6} + \beta_3 D_{t-6} S_{t-6} + w_t \\
&= \begin{cases} \beta_0 + \beta_1 S_{t-6} + w_t & \text{if } S_{t-6} < 0, \\ (\beta_0 + \beta_2) + (\beta_1 + \beta_3) S_{t-6} + w_t & \text{if } S_{t-6} \geq 0 \end{cases}
\end{aligned}$$

```

dummy = ifelse(soi < 0, 0, 1) # for piecewise regression
fish = ts.intersect(rec, soiL6 = stats::lag(soi, -6), dL6 = stats::lag(dummy, -6), dframe=TRUE)
summary(fit <- lm(rec ~ soiL6 * dL6, data = fish, na.action=NULL))
# Coefficients:
# Estimate Std. Error t.value
# (Intercept) 74.479 2.865 25.998
# soiL6 -15.358 7.401 -2.075
# dL6 -1.139 3.711 -0.307
# soiL6:dL6 -51.244 9.523 -5.381
# ---
# Residual standard error: 21.84 on 443 degrees of freedom
# Multiple R-squared: 0.4024
# F-statistic: 99.43 on 3 and 443 DF
attach(fish)
plot(soiL6, rec)
lines(lowess(soiL6, rec), col=4, lwd=2)
points(soiL6, fitted(fit), pch='+', col=2)
plot(resid(fit))
acf(resid(fit)) # not white noise still since lots outside of CI, need to model cycles

```

8.3 Types of Smoothers (MA, Kernel, Lowess, and Splines)

Moving Average Smoothing

If x_t represents the observations, then

$$m_t = \sum_{j=-k}^k a_j x_j \text{ where, } a_j = a_{-j} \geq 0, \text{ and } \sum_{j=-k}^k a_j = 1$$

is a symmetric (two-sided) moving average of the data

Example: Moving Average Smoother on SOI

```

wgts <- c(0.5, rep(1, 11), 0.5)/12 # sums to 1
soif <- stats::filter(soi, sides = 2, filter = wgts)
par()
plot(soi, main="MA smoother of SOI")
lines(soif, lwd = 3, col = "blue")

# To show visually the box-car shape of the weights
par(fig = c(.65, 1, .65, 1), new = TRUE)
nwgts = c(rep(0,20), wgts, rep(0,20))
plot(nwgts, type="l", ylim = c(-.02,.1), xaxt='n', yaxt='n', ann=FALSE)

```

- This method removes (filters out) the obvious annual temperature cycles, and helps visualise the El Nino cycles
- However, it is still quite choppy, probably due to the (relatively non smooth) "boxcar" weights

Kernel Smoothing

If x_t represents the observations, then

$$m_t = \sum_{i=1}^n w_i(t)x_i, \text{ where } w_i(t) = \frac{K(\frac{t-i}{b})}{\sum_{k=1}^n K(\frac{t-k}{b})}$$

are the weights, b is the bandwidth and $K(\cdot)$ is the kernel function.

- Each m_t uses all the x_t 's, contrary to MA smoother.
- The wider the b , the smoother the result.
- A typical kernel is the normal kernel, $K(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}$.
 - In R, `ksmooth(x, y, kernel=c("box", "normal"), bandwidth)`,
 - kernels are scaled such that the kernel quartiles are $\pm 0.25 \times \text{bandwidth}$
 - (in the next example) The `bandwidth` of 1 is approximately smoothing a little over one year over the quartiles

Example: Kernel Smoother on SOI

```
plot(soi)
lines(ksmooth(time(soi), soi, "normal", bandwidth=1), lwd=2, col=4)

# To visually show the gaussian shape of the weights
par(fig = c(.65, 1, .65, 1), new = TRUE)
gauss = function(x) { 1/sqrt(2*pi) * exp(-(x^2)/2) }
x = seq(from = -3, to = 3, by = 0.001)
plot(x, gauss(x), type ="l", ylim=c(-.02,.45), xaxt='n', yaxt='n', ann=FALSE)
```

Lowess Smoothing

- Locally weighted scatter plot smoothers
- Complex, but close to the idea of knn regression, where one uses only the data $(x_{t-k/2}, \dots, x_t, \dots, x_{t+k/2})$ to predict x_t via regression, and then sets $m_t = \hat{x}_t$
- In R, `lowess(x, f=2/3)` where f is the smoother span with default value of $\frac{2}{3}$
- The smoother span is the proportion of points in the plot which influence the smooth at each value, larger f produces more smoothness
- One can also smooth a time series y as a function of another series x via `lowess(x,y)`

Example: Lowess on SOI

```
plot(soi, main="lowess of the SOI series")
lines(lowess(soi, f = 0.05), lwd = 3, col = "blue") # El Nino cycle
lines(lowess(soi), lty = 2, lwd = 3, col = "red") # trend (with default span)
```

Example: Lowess of mortality as a function of temperature

```
plot(temp, cmort, main="Smoothed mortality as a function of temperature",
     xlab="Temp", ylab="Mortality")
lines(lowess(temp, cmort), lwd=3, col="blue")
```

Smoothing Splines

We minimise a compromised between fit and smoothness:

$$\sum_{t=1}^n [x_t - m_t]^2 + \lambda \int (m_t'')^2 dt,$$

where m_t is a cubic spline with a knot at each t . The degree of smoothness is controlled by λ .

- if $\lambda = 0$ then $m_t = x_t$ which is useless and smooths nothing.
- If $\lambda = \infty$ we are infinitely focused on the second derivative of m_t , so that $m_t = c + vt$ which is extremely smooth.
- λ allows thus for a spectrum between linear regression ($\lambda = \infty$) and the data ($\lambda = 0$) — the larger the λ , the smoother the fit.

Example: Splines on SOI

```
plot(soi, main="Smoothing splines fit on the SOI series")
lines(smooth.spline(time(soi), soi, spar = 0.5), lwd = 3, col = "blue") # blue for El Nino cycle
lines(smooth.spline(time(soi), soi, spar = 1), lty = 2, lwd = 3, col = "red") # red for trend
# spar is monotonically related to lambda
```

9 Module 9 - Time Series Models

Overview of main models

Auto regressive models of order p , $AR(p)$:

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + w_t$$

where $\alpha = \mu(1 - \phi_1 - \phi_2 - \dots - \phi_p)$

Moving average model of order q , $MA(q)$:

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \dots + \theta_q w_{t-q}$$

$ARMA(p, q)$ model, combination of the two above:

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q}$$

Other variations:

- $ARIMA(p, d, q)$, which reduces to $ARMA(p, q)$ when differenced d times.
- Multiplicative seasonal ARIMA models:
 - pure seasonal ARMA: $ARMA(P, Q)_s$,
 - multiplicative seasonal ARMA: $ARMA(p, q) \times (P, Q)_s$,
 - seasonal ARIMA: $SARIMA(p, d, q) \times (P, D, Q)_s$.
- Multivariate time series—vector auto-regressive $VAR(p)$ models

9.0.1 Yule-Walker Equations (and PACF)

Partial correlation: $\rho_{XY|Z} = \text{corr}\{X - \hat{X}, Y - \hat{Y}\}$

The partial auto-correlation function (PACF):

For a stationary process, x_t , denoted ϕ_{hh} , for $h = 1, 2, \dots$, is:

$$\phi_{11} = \text{corr}(x_{t+1}, x_t) = \rho(1) \text{ and } \phi_{hh} = \text{cor}(x_{t+h} - \hat{x}_{t+h}, x_t - \hat{x}_t) \text{ for } h \geq 2$$

- the PACF, ϕ_{hh} , is the correlation between x_{t+h} and x_t with the linear dependence of $\{x_{t+1}, \dots, x_{t+h-1}\}$ on each, removed.
- It can be shown that ϕ_{hh} is some coefficient obtained by solving a set of Yule-Walker equations.

Yule-Walker Equations:

Setting $\alpha = 0$ on the centred series x_t :

1. multiply $AR(p)$ and x_{t-k}
2. take expectations
3. utilise the property of $\gamma(-k) = \gamma(k)$
4. divide by auto covariance $\gamma(0)$
5. we obtained a set of $(p+1)$ Yule-Walker equations in the form of $\rho(0) = \phi_1\rho(1) + \phi_2\rho(2) + \dots + \phi_p\rho(p) + \sigma_w^2$, and

$$\rho(1) = \phi_1\rho(0) + \phi_2\rho(1) + \dots + \phi_p\rho(p-1)$$

\vdots

$$\rho(p-1) = \phi_1\rho(p-2) + \phi_2\rho(p-3) + \dots + \phi_p\rho(1)$$

$$\rho(p) = \phi_1\rho(p-1) + \phi_2\rho(p-2) + \dots + \phi_p\rho(0)$$

6. Then the PACF $\phi_{hh} = \phi_h$ is obtained by solving exactly $p = h$ equations.

Properties:

- $\phi_{22} \neq \phi_2$ from solving $p > 2$ equations.
- $\phi_{11} = \phi_1 = \rho(1)$.
- When $h > p$, $x_{t+h} - \hat{x}_{t+h} = w_{t+h}$ which has no overlap with $x_t - \hat{x}_t$ which only depends on $\{w_{t+h-1}, w_{t+h-2}, \dots\} \implies \phi_{hh} = \text{corr}(w_{t+h}, x_t - \hat{x}_t) = 0$, $h > p$.
- The PACF of an $AR(p)$ model will be 0 for $h > p$.

Example Yule-Walker Question:

$$\begin{aligned}\hat{r}(1) &= 0.70, \quad \hat{r}(2) = 0.40, \\ \hat{r}(1) &= \phi_1 + \phi_2 \hat{r}(1), \\ \hat{r}(2) &= \phi_1 \hat{r}(1) + \phi_2, \\ 0.70 &= \phi_1 + 0.70 \phi_2 \implies \phi_1 = 0.70 - 0.70 \phi_2, \\ 0.40 &= 0.70 \phi_1 + \phi_2 = 0.70(0.70 - 0.70 \phi_2) + \phi_2 = 0.49 - 0.49 \phi_2 + \phi_2 = 0.49 + 0.51 \phi_2, \\ \phi_2 &= \frac{0.40 - 0.49}{0.51} \approx -0.1765, \\ \phi_1 &= 0.70 - 0.70(-0.1765) \approx 0.8236.\end{aligned}$$

9.1 AR(p) Models

Of the form

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + w_t,$$

where x_t is stationary, $w_t \sim wn(0, \sigma_w^2)$, and the ϕ_i 's are constants ($\phi_p \neq 0$), and where $\alpha = \mu(1 - \phi_1 - \phi_2 - \dots - \phi_p)$.

If we assume $\alpha = 0$ (which is normal assumption made, unless stated otherwise):

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)x_t = w_t \text{ or } \phi(B)x_t = w_t$$

Example: AR(1) model:

An AR(1) is

$$\begin{aligned}x_t &= \phi x_{t-1} + w_t \\ &= \phi(\phi x_{t-2} + w_{t-1}) + w_t \\ &\vdots \\ &= \phi^k x_{t-k} + \sum_{j=0}^{k-1} \phi^j w_{t-j}\end{aligned}$$

If we continue to iterate backwards (provided $|\phi| < 1$ and $\sup_t Var(x_t) < \infty$):

$$x_t = \sum_{j=0}^{\infty} \phi^j w_{t-j}$$

as a linear process! This is called the stationary solution to the model. Using it we can find:

- Mean: $\mathbb{E}[x_t] = \sum_{j=0}^{\infty} \phi^j \mathbb{E}[w_{t-j}] = 0$,
- Auto-covariance function: $\gamma(h) = \frac{\sigma_w^2 \phi^h}{1 - \phi^2}$, $h \geq 0$

- ACF: $\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \phi^h$, $h \geq 0$
- ACF satisfies the recursion: $\rho(h) = \phi\rho(h-1)$, $h = 1, 2, \dots$

Sample path of AR(1) process ($x_t = 0.9x_{t-1} + w_t$):

```
plot(arima.sim(list(order = c(1,0,0), ar = 0.9), n = 100), ylab = "x")
```

9.1.1 Stationarity and Causality

x_t is both stationary and causal if, and only if, the roots of the characteristic polynomial

$$1 - \phi_1 z - \phi_2 z^2 - \dots - \phi_p z^p = \phi(z) = 0$$

(which can be complex numbers) are all greater than 1 in absolute value (outside the unit circle).

9.1.2 ACF and PACF

For stationary $x_t = \sum_{j=1}^p \phi_j x_{t-j} + w_t$, we have:

$$\gamma(k) = \text{Cov}(\sum_{j=1}^p \phi_j x_{t-j} + w_t, x_{t-k}) = \sum_{j=1}^p \phi_j \gamma(k-j), \text{ for } k \geq 0$$

We can obtain the ACF :

$$\rho(k) = \sum_{j=1}^p \phi_j \rho(k-j), \text{ for } k \geq 0, \text{ since } \rho(h) = \gamma(h)/\gamma(0)$$

We can then write, based on the equation above, the characteristic polynomial

$$\phi(z)\rho(k) = (1 - \phi_1 z - \phi_2 z^2 - \dots - \phi_p z^p)\rho(k) = 0$$

Note:

- For stationary and causal AR, we require $|z_i| > 1$ in order for $\rho(h) = \sum_{j=1}^p A_j z_j^{-h} \rightarrow 0$ as $h \rightarrow \infty$
- If all roots are real, then $\rho(h)$ dampens exponentially to zero as $h \rightarrow \infty$
- If some of the roots are complex, then $\rho(h)$ will dampen, in a sinusoidal fashion, exponentially to zero as $h \rightarrow \infty$
- The exponential decay to zero property flows on to ARMA models

PACF:

- For $AR(p)$ model, $\phi_{hh} = 0$ for $h > p$.
- Earlier values of $\phi_{11}, \phi_{22}, \dots, \phi_{pp}$ are not necessarily 0.
- The cutting off of ϕ_{hh} after p lags is the signature of the $AR(p)$ model.

Example: ACF of $x_t = w_t - 0.7x_{t-1}$:

$$\begin{aligned} \gamma(k) &= (x_t, x_{t-k}) = (w_t - 0.7x_{t-1}, x_{t-k}) \\ &= (w_t - 0.7(w_{t-1} - 0.7x_{t-2}), x_{t-k}) \\ &= (w_t - 0.7(w_{t-1} - 0.7(\dots - 0.7(w_{t-k+1} - 0.7x_{t-k})))), x_{t-k}) \\ &\stackrel{*}{=} (-0.7(-0.7(\dots - 0.7(-0.7x_{t-k})))), x_{t-k}) \\ &= (-0.7)^k (x_{t-k}, x_{t-k}) = (-0.7)^k \gamma(0) \end{aligned}$$

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)} = \frac{(-0.7)^k \gamma(0)}{\gamma(0)} = (-0.7)^k, \quad k \geq 0.$$

* using $(w_{t-k+s}, x_{t-k}) = 0$ for $s \geq 1$.

Examples of AR ACFs and PACFs in R:

```
set.seed(2)
AR1 = arima.sim(list(order=c(2,0,0), ar=c(0.8,-0.1)), n=10000)
par(mfrow=c(1,2))
acf(AR1)
pacf(AR1) # display sample of phi_{hh} for h > 0

# OR can just use acf2 to show both ACF and PACF
acf2(AR1) # pacf plot doesnt show value 1 at lag 0
```

9.1.3 Explosive AR models and Causality

When a process does not depend on the future – such as AR(1) when $|\phi| < 1$ – we will say that the process is causal.

In general for $AR(p)$ where $p > 1$:

- Stationary and causal are not equivalent conditions, `stationary` \neq `causal`.
- You may have a AR model will future dependency without knowing it, so it would be not causal.
- The condition for $|z| > 1$ for $\phi(z) = 0$ stated "stationary and causal".

Example with AR(1) of stationary but not causal:

Assume for $AR(1)$ that $|\phi| > 1$, write (by iterating forward k steps):

$$\begin{aligned} x_{t+1} &= \phi x_t = w_{t+1} \\ x_t &= \phi^{-1} x_{t+1} - \phi^{-1} w_{t+1} \\ &= \phi^{-1}(\phi^{-1} x_{t+2} - \phi^{-1} w_{t+2}) - \phi^{-1} w_{t+1} \\ &\vdots \\ &= \phi^{-k} x_{t+k} - \sum_{j=1}^{k-1} \phi^{-j} w_{t+j} \text{ (future dependence)} \end{aligned}$$

Since we assumed $|\phi| > 1 \implies |\phi|^{-1} < 1$, suggesting that the future dependent $AR(1)$ model is stationary, and is therefore not causal.

9.2 MA(q) models

Of the form

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \dots + \theta_q w_{t-q} = \theta(B) w_t$$

where $w_t \sim wn(0, \sigma_w^2)$ and $\theta_q \neq 0$. Unlike AR, the MA is stationary for ANY value of parameters θ_i .

Example: MA(1) model:

An MA(1) is

$$x_t = w_t + \theta w_{t-1}$$

hence the mean $\mathbb{E}[x_t] = 0$ and the auto covariance function:

$$\gamma(h) = \begin{cases} (1 + \theta^2)\sigma_w^2 & h = 0, \\ \theta\sigma_w^2 & h = 1, \\ 0 & h > 1 \end{cases}$$

and the ACF is (furthermore $|\rho(1)| \leq 1/2, \forall \theta$)

$$\rho(h) = \begin{cases} 1 & h = 0, \\ \frac{\theta}{1+\theta^2} & h = 1, \\ 0 & h > 1 \end{cases}$$

Sample path of MA(1) process ($x_t = w_t + 0.5w_{t-1}$):

```
plot(arima.sim(list(order = c(0,0,1), ma=0.5), n=100), ylab = "x")
```

9.2.1 Non-uniqueness and Invertibility

Non-uniqueness:

As seen in the tute for a MA(1) process the following two series give the same $\gamma(h)$ and $\rho(h)$: $x_t = w_t + \frac{1}{5}w_{t-1}$, $w_t \sim N(0, 25)$ and $y_t = v_t + 5v_{t-1}$, $v_t \sim N(0, 1)$. There is a "correct" one to choose and we choose the one that is invertible.

Invertibility:

Example for the case above:

If we invert the roles of x_t and w_t in x_t MA(1) as above:

$$\begin{aligned} w_t &= -\theta w_{t-1} + x_t \\ &= -\theta(-\theta w_{t-2} + x_{t-1}) + x_t \\ &= \sum_{j=0}^{\infty} (-\theta)^j x_{t-j} \quad \text{if } |\theta| < 1 \end{aligned}$$

which is an infinite AR representation of the model. Since we need $|\theta| < 1$ for this to work (for the series to converge), we need to choose the MA(1) model with $\theta = 1/5, \sigma_w^2 = 25$.

General Case ($q > 1$):

$$x_t = \theta(B)w_t \Leftrightarrow \pi(B)x_t = w_t, \text{ where } \pi(B) = \theta^{-1}(B)$$

Just as we required $|\theta| < 1$ for invertibility, we will require **the roots of $\theta(z)$ to lie outside the unit circle**. ($\theta(z) = 0$ only when $|z| > 1$).

General case applied to MA(1):

$$\theta(z) = 1 + \theta z \Leftrightarrow \pi(z) = \theta^{-1}(z) = \frac{1}{1 + \theta z} = \sum_{j=0}^{\infty} (-\theta)^j z^j \text{ if } |\theta z| < 1$$

Consequently, we can write: $\pi(B) = \sum_{j=0}^{\infty} (-\theta^j)B^j$

9.2.2 ACF and PACF

We have

$$\begin{aligned}\gamma(h) &= \text{Cov}(x_{t+h}, x_t) \\ &= \text{Cov}\left(\sum_{j=0}^q \theta_j w_{t+h-j}, \sum_{k=0}^q \theta_k w_{t-k}\right) \\ &= \begin{cases} \sigma_w^2 \sum_{j=0}^{q-h} \theta_j \theta_{j+h} & 0 \leq h \leq q, \\ 0 & h > q \end{cases} \\ &= \gamma(-h)\end{aligned}$$

The ACF is then

$$\rho(h) = \begin{cases} \frac{\sum_{j=0}^{q-h} \theta_j \theta_{j+h}}{1 + \theta_1^2 + \dots + \theta_q^2} & 1 \leq h \leq q \\ 0 & h > q \end{cases}$$

The cutting off of $\gamma(h)$ at q lags is the signature of the $MA(q)$ model.

An invertible $MA(q)$ can be written as

$$x_t = -\sum_{j=1}^{\infty} \pi_j x_{t-j} + w_t$$

No finite representation of the PACF exists, and hence the PACF will never cut off, as opposed to the case of $AR(p)$. Instead the PACF will decay exponentially to 0.

Examples of MA ACFs and PACFs in R:

```
set.seed(2)
MA1 = arima.sim(list(order=c(0,0,2), ma=c(-0.8, -0.1)), n=10000)
par(mfrow=c(1,2))
acf(MA1)
pacf(MA1)
```

9.3 ARMA(p, q) models

Is stationary and is of form

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q} \Leftrightarrow \phi(B)x_t = \theta(B)w_t$$

with $\phi_p \neq 0$, $\theta_q \neq 0$, $w_t \sim WN(0, \sigma_w^2)$. If x_t has non-zero μ , we set $\alpha = \mu(1 - \phi_1 - \dots - \phi_p)$ and add it into RHS of the equation.

9.3.1 Parameter redundancy

If there is a common factor between $\phi(B)$ and $\theta(B)$ that we can cancel out then those parameters are redundant and our model can become simpler. Therefore, we must cancel any common factors present in our question.

9.3.2 Causality and (non)-future dependence

$ARMA(p, q)$ model is said to be causal, if we can write (so its a one-sided linear process):

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} = \psi(B)w_t$$

Where $\psi(B) = \sum_{j=0}^{\infty} \psi_j B^j$, and $\sum_{j=0}^{\infty} |\psi_j| < \infty$; we set $\psi_0 = 1$.

When is it causal?

Causal if and only if $\phi(z) = 0$ only if $|z| > 1$. In other words the **roots of $\phi(z)$ must lie outside the unit circle**.

Since, the coefficients of the linear process given above can be determined by solving

$$\psi(z) = \sum_{j=0}^{\infty} \psi_j z^j = \frac{\theta(z)}{\phi(z)}, |z| < 1$$

9.3.3 Invertibility and non-uniqueness

$ARMA(p, q)$ model is said to be invertible, if we can write:

$$\pi(B)x_t = \sum_{j=0}^{\infty} \pi_j x_{t-j} = w_t$$

Where $\pi(B) = \sum_{j=0}^{\infty} \pi_j B^j$, and $\sum_{j=0}^{\infty} |\pi_j| < \infty$; we set $\pi_0 = 1$.

When is it invertible?

Invertible if and only if $\theta(z) = 0$ only if $|z| > 1$. In other words the **roots of $\theta(z)$ lie outside the unit circle**.

Since, the coefficients of π_j of $\pi(B)$ given above can be determined by solving

$$\pi(z) = \sum_{j=0}^{\infty} \pi_j z^j = \frac{\phi(z)}{\theta(z)}, |z| \leq 1$$

Example: parameter redundancy, Causality, Invertibility

Consider the process:

$$x_t = 0.4x_{t-1} + 0.45x_{t-2} + w_t + w_{t-1} + 0.25w_{t-2}$$

There is a common factor of $(1 - 0.5B)$ between $\phi(B)$ and $\theta(B)$ so we can reduce the parameters by cancelling the common factors on both sides. So the model is now in fact:

$$x_t = 0.9x_{t-1} + 0.5w_{t-1} + w_t, \text{ with } \phi(B) = 1 - 0.9B \text{ and } \theta(B) = 1 + 0.5B$$

Causal check

Let $\phi(z) = 1 - 0.9z = 0 \implies z = 10/9 > 1 \implies$ causal

Invertibility check

Let $\theta(z) = 1 + 0.5z = 0 \implies z = -2$, since $|z| > 1 \implies$ invertible

Find linear representation to find weights ψ

$$\begin{aligned}\phi(z)\psi(z) &= \theta(z) \\ (1 - 0.9z)(1 + \psi_1 z + \psi_2 z^2 + \dots + \psi_j z^j + \dots) &= 1 + 0.5z \\ (\text{regrouping coefficients of power } z) \\ 1 + (\psi_1 - 0.9)z + (\psi_2 - 0.9\psi_1)z^2 + \dots + (\psi_j - 0.9\psi_{j-1})z^j + \dots &= 1 + 0.5z\end{aligned}$$

Note that all coefficients of z^j are 0 for $j > 1$ on the RHS. We obtain then

$$\begin{aligned}\psi_1 - 0.9 &= 0.5 \implies \psi_1 = 1.4, \\ \psi_j - 0.9\psi_{j-1} &= 0 \implies \frac{\psi_j}{\psi_{j-1}} = 0.9, \quad j > 1 \\ \psi_j &= 0.9\psi_{j-1} = (0.9)^2\psi_{j-2} = \dots = (0.9)^{j-1}\psi_1\end{aligned}$$

and thus

$$\psi_j = 1.4(0.9)^{j-1} \text{ for } j \geq 1$$

and hence

$$x_t = w_t + 1.4 \sum_{j=1}^{\infty} 0.9^{j-1} w_{t-j}$$

In R, we just use $(x_t = 0.9x_{t-1} + 0.5w_{t-1} + w_t)$

```
format(ARMAtoMA(ar=0.9, ma=0.5, 10), digits=2) # first 10 psi-weights
## [1] "1.40" "1.26" "1.13" "1.02" "0.92" "0.83" "0.74" "0.67" "0.60"
## [10] "0.54"
```

Find invertible representation to find weights π

$$\begin{aligned}\theta(z)\pi(z) &= \phi(z) \\ (1 + 0.5z)(1 + \pi_1 z + \pi_2 z^2 + \pi_3 z^3 + \dots) &= 1 - 0.9z \\ (\text{regrouping coefficients of power } z) \\ 1 + (\pi_1 + 0.5)z + (\pi_2 + 0.5\pi_1)z^2 + \dots + (\pi_j + 0.5\pi_{j-1})z^j + \dots &= 1 - 0.9z\end{aligned}$$

By equating coefficients we get

$$\begin{aligned}\pi_1 + 0.5 &= -0.9 \implies \pi_1 = -1.4 \\ \pi_j + 0.5\pi_{j-1} &= 0 \implies \pi_j = -0.5\pi_{j-1} \\ \pi_j &= 0.5^2\pi_{j-2} = -0.5^3\pi_{j-3} = \dots = (-1)^{j-1}(0.5)^{j-1}\pi_0 \\ \pi_j &= (-1)^j 1.4(0.5)^{j-1} \text{ for } j \geq 1\end{aligned}$$

and then

$$w_t = \sum_{j=0}^{\infty} \pi_j x_{t-j} = x_t + \sum_{j=1}^{\infty} \pi_j x_{t-j}$$

so that

$$x_t = - \sum_{j=1}^{\infty} \pi_j x_{t-j} + w_t$$

In R, we just use $(w_t = -0.5w_{t-1} - 0.9x_{t-1} + x_t)$

```
format(ARMAtoMA(ar=-0.5, ma=-0.9, 10), digits = 1) # first 10 pi-weights
## [1] "-1.400" "0.700" "-0.350" "0.175" "-0.087" "0.044" "-0.022"
## [8] "0.011" "-0.005" "0.003"
```

to be stationary the roots just need to not be on the unit circle.

Revisiting the future-dependent example

We have: $x_t = \phi x_{t-1} + w_t$, so the characteristic equation is $\phi(z) = 1 - \phi z$, which has the root $z_0 = \frac{1}{\phi}$.

There are three cases:

1. $\phi < 1$: the root is not on the unit circle, and also outside the unit circle, so that the process—which is AR(1)—is stationary and causal.
2. $\phi = 1$: the root is on the unit circle, and hence the process is not stationary. Note it is not causal either because this linear process depends on the past and present (this is the random walk case), it does not satisfy the requirement for it to be a linear process.
3. $\phi > 1$: the root is not on the unit circle, and hence the process is stationary. However, the root is inside the unit circle, which implies it is not causal. The process will depend on the future as demonstrated earlier.

9.3.4 ACF

For a causal ARMA(p, q) model: $\phi(B)x_t = \theta(B)w_t$ we use the linear representation $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$. It follows immediately that $\mathbb{E}[x_t] = 0$, and the autocovariance function of x_t is:

$$\gamma(h) = \text{Cov}(x_{t+h}, x_t) = \sigma_w^2 \sum_{j=0}^{\infty} \psi_j \psi_{j+h}, \quad h \geq 0$$

This approach requires us to solve for the ψ 's.

Alternatively it is possible to write a general homogeneous equation for the ACF of a causal ARMA process to solve for the γ 's directly:

$$\gamma(h) - \phi_1 \gamma(h-1) - \dots - \phi_p \gamma(h-p) = 0, \quad h \geq \max(p, q+1)$$

with initial conditions

$$\gamma(h) - \sum_{j=1}^p \phi_j \gamma(h-j) = \sigma_w^2 \sum_{j=h}^q \theta_j \psi_{j-h}, \quad 0 \leq h < \max(p, q+1)$$

Finally, the ACF is

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)}$$

In general, the ACF cannot distinguish between AR and ARMA, which is why PACF is useful (in presence of pure AR it will cut off, for ARMA it will tail off instead).

Example: ACF of ARMA(1,1):

$$x_t = \phi x_{t-1} + \theta w_{t-1} + w_t, \quad \text{where } |\phi| < 1$$

The auto-covariance then satisfies

$$\gamma(h) - \phi \gamma(h-1) = 0, \quad h = 2, 3, \dots$$

which has general solution

$$\gamma(h) = c\phi^h, \quad h = 1, 2, \dots$$

Initial conditions are

$$\begin{aligned} \gamma(0) &= \phi \gamma(1) + \sigma_w^2 (\theta_0 \psi_0 + \theta_1 \psi_1) = \phi \gamma(1) + \sigma_w^2 [1 + \theta \phi + \theta^2] \\ \gamma(1) &= \phi \gamma(0) + \sigma_w^2 \theta \end{aligned}$$

Note that $\psi_1 = \theta + \phi$ for an $ARMA(1, 1)$ model. Then solving for $\gamma(0)$ and $\gamma(1)$ yields

$$\begin{aligned}\gamma(0) &= \sigma_w^2 \frac{1 + 2\theta\phi + \theta^2}{1 - \phi^2} \\ \gamma(1) &= \sigma_w^2 \frac{(1 + \theta\phi)(\phi + \theta)}{1 - \phi^2}\end{aligned}$$

Since $\gamma(1) = c\phi$, we get $c = \gamma(1)/\psi$ and

$$\gamma(h) = \frac{\gamma(1)}{\phi} \phi^h = \sigma_w^2 \frac{(1 + \theta\phi)(\phi + \theta)}{1 - \phi^2} \phi^{h-1}, \quad h \geq 1$$

from which we obtain the ACF

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \frac{\gamma(1)}{\gamma(0)} \phi^{h-1} = \frac{(1 + \theta\phi)(\phi + \theta)}{1 + 2\theta\phi + \theta^2} \phi^{h-1}, \quad h \geq 1$$

this has the same pattern as $AR(1)$.

9.4 ARIMA(p, d, q) models

Of the form

$$\phi(B)(1 - B)^d x_t = \theta(B)w_t, \quad \text{where } \nabla^d x_t = (1 - B)^d x_t$$

If $\mathbb{E}[\nabla^d x_t] = \mu_t$ we add δ to the RHS of the equation above, where $\delta = \mu(1 - \phi_1 - \dots - \phi_p)$.

9.5 Multiplicative Seasonal ARIMA models

Previously we have observed seasonality in many time series so far. For example for a seasonal lag s of economic data ($s = 12$) or quarterly data ($s = 4$).

Seasonality can be due to:

- Deterministic reasons: include in trend
- Stochastic dependence at seasonal lags s : use SARIMA models

9.5.1 Pure seasonal models: $ARMA(P, Q)_s$

Of the form

$$\Phi_P(B^s)x_t = \Theta_Q(B^s)w_t$$

Here the inter-temporal correlations are exclusively seasonal (there are no non-seasonal correlations).

Note (*analogous to nonseasonal ARMA*):

- It will be **causal** only when the roots of $\Phi_P(z^s)$ lie **outside of the unit circle**.
- It will be **invertible** only when the roots of $\Theta_Q(z^s)$ lie **outside of the unit circle**.

Example: first order seasonal ($s=12$) AR model:

$s = 12$, $P = 1$, $Q = 0$, and it can be written as

$$(1 - \Phi B^{12})x_t = w_t \text{ or } x_t = \Phi x_{t-12} + w_t$$

Note:

- x_t is expressed in terms of past lags at multiples of the (yearly) seasonal period $s = 12$ months,
- very similar to the unit lag model of $AR(1)$ that we know,

- causal if $|\Phi| < 1$

Simulated example (with $\Phi = 0.9$) and plot time series:

```
sed.seed(2)
Phi = c(rep(0, 11), 0.9) # all Phi from 0 to 11 is 0 except for 12
sAR = arima.sim(list(order = c(12, 0, 0), ar = Phi), n=37)
sAR = ts(sAR, freq = 12)

# Plot
par(mar = c(3, 3, 2, 1), mgp = c(1.6, 0.6, 0))
plot(sAR, axes=FALSE, main="Seasonal AR(1)", xlab = "year", type = "c")
Months <- c("J", "F", "M", "A", "M", "J", "J", "A", "S", "O", "N", "D")
points(sAR, pch = Months, cex = 1.25, font = 4, col = 1:4)
axis(1, 1:4)
abline(v = 1:4, lty = 2, col = gray(0.7))
axis(2)
box()
```

Theoretical ACF and PACF of the model (and plot):

```
Phi = c(rep(0, 11), 0.9)
ACF = ARMAacf(ar = Phi, ma = 0, 100) # can do [-1] on end to remove lag at 0
PACF = ARMAacf(ar = Phi, ma = 0, 100, pacf = TRUE) # need pacf argument
par(mfrow = c(2,1), mar = c(3,3,2,1), mgp=c(1.6,0.6,0))
plot(ACF, type="h", xlab="LAG", ylim=c(-1, 1)) # tails off (lines shifted by 1 unit to the right)
abline(h=0)
plot(PACF, type="h", xlab="LAG", ylim=c(-0.1,1)) # 1 spike at 12 as expected
abline(h=0)
```

ACF of first-order seasonal MA model $s = 12$:

$$x_t = w_t + \Theta w_{t-12}$$

we have

$$\begin{aligned}\gamma(0) &= (1 + \Theta^2)\sigma^2 \\ \gamma(\pm 12) &= \Theta\sigma^2 \\ \gamma(h) &= 0, \text{ otherwise.}\end{aligned}$$

Thus, the only nonzero correlation (aside from lag zero) is

$$\rho(\pm 12) = \frac{\Theta}{1 + \Theta^2}$$

The PACF will tail off at multiples of $s = 12$.

ACF of first-order seasonal AR model $s = 12$:

$$x_t = \Phi x_{t-12} + w_t$$

we have

$$\begin{aligned}\gamma(0) &= \frac{\sigma^2}{1 - \Phi^2} \\ \gamma(\pm 12k) &= \frac{\sigma^2 \Phi^k}{1 - \Phi^2}, \quad k = 1, 2, \dots \\ \gamma(h) &= 0, \text{ otherwise.}\end{aligned}$$

Thus, the only nonzero correlation (aside from lag zero) are

$$\rho(\pm 12) = \Phi^k, k = 1, 2, \dots$$

The PACF will have one nonzero correlation at $s = 12$ and then cut off.

9.5.2 Multiplicative seasonal models: ARMA(p, q) \times (P, Q)_s

Of the form

$$\Phi_P(B^s)\phi(B)x_t = \Theta_Q(B^s)\theta(B)w_t$$

Note:

- When selecting a model, we need to carefully examine the ACF and PACF of the data,
- Choosing this model first will generally lead to better results.

Example: a mixed seasonal model $ARMA(0, 1) \times (1, 0)_{12}$:

$$x_t = \Phi x_{t-12} + w_t + \theta w_{t-1}$$

where $|\Phi| < 1$ and $|\theta| < 1$. Because

- x_{t-12}, w_t , and w_{t-1} are uncorrelated; and
- x_t is stationary

Then

$$\gamma(0) = \Phi^2\gamma(0) + \sigma_w^2 + \theta^2\sigma_w^2 \Leftrightarrow \gamma(0) = \frac{1 + \theta^2}{1 - \Phi^2}\sigma_w^2$$

Furthermore, multiplying the model by x_{t-h} , $h > 0$

$$x_t x_{t-h} = \Phi x_{t-12} x_{t-h} + w_t x_{t-h} + \theta w_{t-1} x_{t-h}$$

and taking expectations leads to

$$\begin{aligned} \mathbb{E}[x_t x_{t-1}] &= \gamma(1) = \mathbb{E}[(\Phi x_{t-12} + w_t + \theta w_{t-1})x_{t-1}] \\ &= \Phi\mathbb{E}[x_{t-12}x_{t-1}] + \mathbb{E}[w_t x_{t-1}] + \theta\mathbb{E}[w_{t-1}x_{t-1}] \\ &= \Phi\gamma(11) + 0 + \theta\sigma_w^2 \\ &= \Phi\gamma(11) + \theta\sigma_w^2 \end{aligned}$$

$$\mathbb{E}[x_t x_{t-12}] = \gamma(12) = \Phi\gamma(h-12), \text{ for } h \geq 2 \text{ (using similar proof)}$$

Thus, the ACF for this model (requires some algebra) is

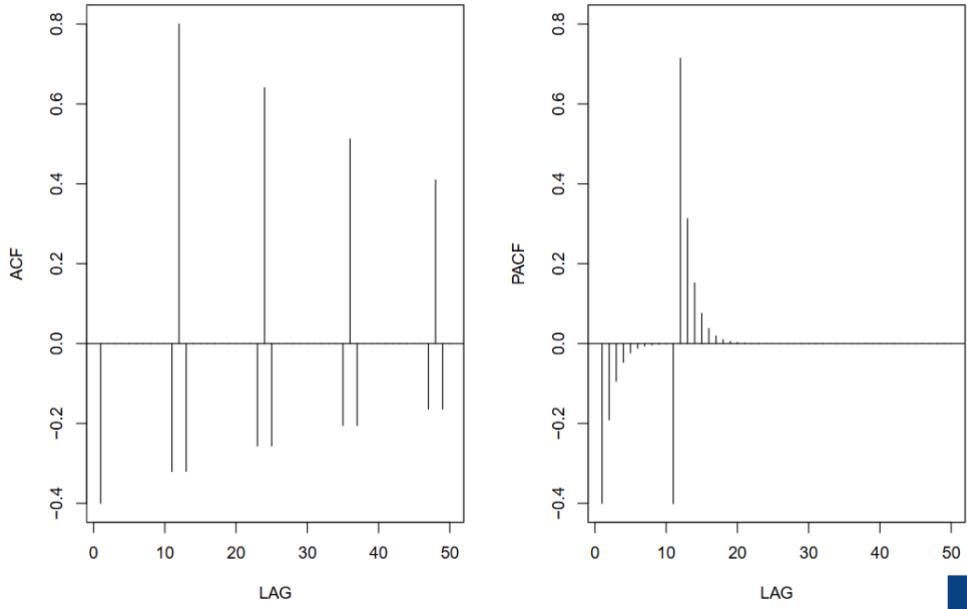
$$\begin{aligned} \rho(12h) &= \Phi^h, h = 1, 2, \dots \\ \rho(12h-1) &= \rho(12h+1) = \frac{\theta}{1+\theta^2}\Phi^h, h = 0, 1, 2, \dots \\ \rho(h) &= 0, \text{ otherwise.} \end{aligned}$$

Example: if $\Phi = 0.8$ and $\theta = -0.5$, then the theoretical ACF and PACF become

```
Phi = c(rep(0,11), 0.8)
ACF = ARMAacf(ar = Phi, ma = -0.5, 50)[-1] # [-1] removes lag at time 0
PACF = ARMAacf(ar = Phi, ma = -0.5, 50, pacf = TRUE)
# Plotting
par(mfrow = c(1,2))
plot(ACF, type = "h", xlab = "LAG", ylim = c(-0.4, 0.8))
abline(h=0)
plot(PACF, type = "h", xlab = "LAG", ylim = c(-0.4, 0.8))
abline(h=0)
```

Output and notes on ACF and PACF graph:

- ACF:
 - Slowly, tailing off towards 0
 - From the non-seasonal $MA(1)$ part, we have the negative values either side of the positive values at lags of ± 1 of lags of multiples of 12 from the seasonality of the $sAR(1)_{12}$
- PACF:
 - Cut off can be seen after lag 12 $\implies sAR(1)_{12}$
 - Tails off for non-seasonal component reinforcing the MA part of the model



9.6 SARIMA model $ARIMA(p, d, q) \times (P, D, Q)_s$

Seasonal differencing:

- Can be indicated when the ACF decays slowly at multiples of some season s , but negligible between the seasons.
- Seasonal difference of order D means $\nabla_s^D x_t = (1 - B^s)^D x_t$.
- Typically, $D = 1$ is sufficient to obtain seasonal stationarity

SARIMA Model:

Of the form

$$\Phi_P(B^s)\phi(B)\nabla_s^D\nabla^d x_t = \delta + \Theta_Q(B^s)\theta(B)w_t$$

Note:

- Denoted as $ARIMA(p, d, q) \times (P, D, Q)_s$.
- Ordinary difference: $\nabla^d = (1 - B)^d$ and seasonal difference: $\nabla_s^D = (1 - B^s)^D$
- Non-seasonal MA component has polynomial $\theta(B)$ with order q . Non seasonal AR component has polynomial $\phi(B)$ with order p .
- Seasonal MA component has polynomial $\Theta(B)$ with order Q . Seasonal AR component has polynomial $\Phi(B)$ with order P .

Example: a typical SARIMA model $ARIMA(0, 1, 1) \times (0, 1, 1)_{12}$:

We set $\delta = 0$: $\nabla_{12} \nabla x_t = \Theta(B^{12})\theta(B)w_t$. Can also be represented as:

$$(1 - B^{12})(1 - B)x_t = (1 + \Theta B^{12})(1 + \theta B)w_t$$

$$(1 - B - B^{12} + B^{13})x_t = (1 + \theta B + \Theta B^{12} + \Theta \theta B^{13})w_t$$

$$x_t = x_{t-1} + x_{t-12} - x_{t-13} + w_t + \theta w_{t-1} + \Theta w_{t-12} + \Theta \theta w_{t-13}$$

Example: Air Passengers dataset

```
x = AirPassengers
plot.ts(x, main = "Air Passengers series, unmodified")
# shows trend and increasing variance so should try log transformation

lx = log(x)
plot.ts(lx, main = "Log of Air Passenger series") # this stabilised the variance

dlx = diff(lx) # try differencing to remove the trend
plot.ts(dlx, main = "Differenced Log of Air Passenger series")
# There is still persistence in spikes with the seasons dlx_t approx. equal dlx_{t-12}

ddlx = diff(dlx, 12) # apply twelfth-order difference on differenced data
plot.ts(ddlx) # improved the seasonality

# Aggregating graphs even more not more analysis though
par(mfrow=c(2,1), mar=c(3,3,2,1), mgp=c(1.6,0.6,0))
monthplot(dlx) # mean on monthly basis varies a lot
monthplot(ddlx) # mean on a monthly basis is very similar
plot.ts(cbind(x, lx, dlx, ddlx), main = "") # shows all plots together
```

9.7 Multivariate time series - VAR(p) models

- Many data sets involve more than one time series, and we are often interested in the possible dynamics relating all series.
- Thus we are interested in modelling and forecasting $k \times 1$ vector-valued time series:

$$x_t = (x_{t1}, \dots, x_{tk})', t = 0, \pm 1, \pm 2, \dots$$

- The *univariate* AR model can be extended to the *Vector Auto-Regressive* (VAR) model.

9.7.1 VAR(1) model

Of the form

$$x_t = \alpha + \phi x_{t-1} + w_t,$$

Where,

- ϕ is a $k \times k$ transition matrix, that expresses the dependence of x_t on x_{t-1} (not these are vectors).
- The vector white noise process w_t is assumed to be multivariate normal with mean-zero and covariance matrix $\mathbb{E}[w_t w_t'] = \Sigma_w$.
- The vector $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)'$ is similar to a constant in the regression setting. If $\mathbb{E}[x_t] = \mu$, then $\alpha = (\mathbf{I} - \phi)\mu$, as before.

Example: Mortality, Temperature, Pollution:

- Define: $x_t = (x_{t1}, x_{t2}, x_{t3})'$, as a vector of dimension $k = 3$ for cardiovascular mortality x_{t1} , temperature x_{t2} , and particulate levels x_{t3} .

- We might envision dynamic relations with first order relation

$$\begin{aligned}x_{t1} &= \alpha_1 + \beta_1 t + \phi_{11}x_{t-1,1} + \phi_{12}x_{t-1,2} + \phi_{13}x_{t-1,3} + w_{t1} \\x_{t2} &= \alpha_2 + \beta_2 t + \phi_{21}x_{t-1,1} + \phi_{22}x_{t-1,2} + \phi_{23}x_{t-1,3} + w_{t2} \\x_{t3} &= \alpha_3 + \beta_3 t + \phi_{31}x_{t-1,2} + \phi_{32}x_{t-1,3} + \phi_{33}x_{t-1,3} + w_{t3}\end{aligned}$$

- Also can be written in matrix form as

$$x_t = \Gamma u_t + \phi x_{t-1} + w_t$$

– Where $\Gamma = [\alpha|\beta]$ is 3×2 and $u_t = (1, t)'$ is 2×1

We use R package `vars` to fit VAR models via last squares.

```
library(vars)
x = cbind(cmort, tempr, part) # make a vector of datasets
summary(VAR(x, p=1, type="both")) # p=1 for Var(1), "both" fits constant + trend

# VAR Estimation Results:
# =====
# Endogenous variables: cmort, tempr, part
# Deterministic variables: both
# Sample size: 507
# Log Likelihood: -5116.02
# Roots of the characteristic polynomial:
# 0.8931 0.4953 0.1444 <- IN VAR WE WANT THESE |z| < 1. Good!
# Call:
# VAR(y = x, p = 1, type = "both")

# Estimation results for equation cmort:
# =====
# cmort = cmort.l1 + tempr.l1 + part.l1 + const + trend
#
#           Estimate Std. Error t value Pr(>|t|)
# cmort.l1  0.464824  0.036729 12.656 < 2e-16 ***
# tempr.l1 -0.360888  0.032188 -11.212 < 2e-16 ***
# part.l1   0.099415  0.019178  5.184 3.16e-07 ***
# const     73.227292  4.834004 15.148 < 2e-16 ***
# trend    -0.014459  0.001978 -7.308 1.07e-12 ***
# ---
# Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
#
#
# Residual standard error: 5.583 on 502 degrees of freedom
# Multiple R-Squared: 0.6908,   Adjusted R-squared: 0.6883
# F-statistic: 280.3 on 4 and 502 DF,  p-value: < 2.2e-16

# Estimation results for equation tempr:
# =====
# tempr = cmort.l1 + tempr.l1 + part.l1 + const + trend
#
#           Estimate Std. Error t value Pr(>|t|)
# cmort.l1 -0.244046  0.042105 -5.796 1.20e-08 ***
# tempr.l1  0.486596  0.036899 13.187 < 2e-16 ***
# part.l1  -0.127661  0.021985 -5.807 1.13e-08 ***
# const    67.585598  5.541550 12.196 < 2e-16 ***
# trend    -0.006912  0.002268 -3.048  0.00243 **
# ---
```

```

# Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
#
#
# Residual standard error: 6.4 on 502 degrees of freedom
# Multiple R-Squared: 0.5007,   Adjusted R-squared: 0.4967
# F-statistic: 125.9 on 4 and 502 DF,  p-value: < 2.2e-16

# Estimation results for equation part:
# =====
# part = cmort.l1 + tempr.l1 + part.l1 + const + trend
#
#           Estimate Std. Error t value Pr(>|t|)
# cmort.l1 -0.124775  0.079013 -1.579   0.115 <- MORTALITY NOT SIG. TO PREDICT POLLUTION
# tempr.l1 -0.476526  0.069245 -6.882 1.77e-11 ***
# part.l1   0.581308  0.041257 14.090 < 2e-16 ***
# const     67.463501 10.399163  6.487 2.10e-10 ***
# trend    -0.004650  0.004256 -1.093   0.275
# ---
# Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
#
#
# Residual standard error: 12.01 on 502 degrees of freedom
# Multiple R-Squared: 0.3732,   Adjusted R-squared: 0.3683
# F-statistic: 74.74 on 4 and 502 DF,  p-value: < 2.2e-16

# Covariance matrix of residuals:
#      cmort  tempr  part
# cmort 31.172  5.975 16.65
# tempr  5.975 40.965 42.32
# part   16.654 42.323 144.26
#
# Correlation matrix of residuals:
#      cmort  tempr  part
# cmort 1.0000 0.1672 0.2484
# tempr  0.1672 1.0000 0.5506
# part   0.2484 0.5506 1.0000

```

By inspecting the top triangle of the correlation matrix of residuals the values are pretty large suggesting not white noise (note: this is not a formal way to test this!)

9.7.2 $\text{Var}(p)$ models

- It is easy to extend $\text{Var}(1)$ process to higher orders (with correlations going farther than one step into the past), leading to $\text{Var}(p)$.
- The regressors are $(1, x'_{t-1}, x'_{t-2}, \dots, x'_{t-p})'$.
- The regression model is then

$$x_t = \alpha + \sum_{j=1}^p \phi_j x_{t-j} + w_t$$

- The R function `VARselect` suggests the optimal order p according to different criteria:
 - AIC,
 - Hannan-Quinn (similar to BIC),
 - BIC (called $SC(n)$ in the R function output),
 - Final Prediction Error (minimises the approximate mean squared one-step-ahead prediction error).

Example: $VAR(p)$ on Mortality, Temperature, Pollution:

First look for optimal p order value. Note that the **smaller** the order the better as we always prefer a smaller model than a larger one.

```
library(vars)
VARselect(x, lag.max = 10, type = "both")
# $selection
# AIC(n)  HQ(n)  SC(n)  FPE(n)
#      9      5      2      9 <- 9 IS GOOD BUT WE PREFER SIMPLER 2 MODEL
#
# $criteria
#          1      2      3      4      5
# AIC(n) 11.73780 11.30185 11.26788 11.23030 11.17634
# HQ(n)  11.78758 11.38149 11.37738 11.36967 11.34557
# SC(n)  11.86463 11.50477 11.54689 11.58541 11.60755
# FPE(n) 125216.91717 80972.28678 78268.19568 75383.73647 71426.10041
#          6      7      8      9      10
# AIC(n) 11.15266 11.15247 11.12878 11.11915 11.12019
# HQ(n)  11.35176 11.38144 11.38760 11.40784 11.43874
# SC(n)  11.65996 11.73587 11.78827 11.85473 11.93187
# FPE(n) 69758.25113 69749.89175 68122.40518 67476.96374 67556.45243
```

So we will proceed with $p = 2$ according to BIC .

```
library(vars)
summary(fit <- VAR(x, p = 2, type = "both"))

# VAR Estimation Results:
# =====
# Endogenous variables: cmort, tempr, part
# Deterministic variables: both
# Sample size: 506
# Log Likelihood: -4987.186
# Roots of the characteristic polynomial:
# 0.8807 0.8807 0.5466 0.4746 0.4746 0.4498 <- STILL < 1 SO GOOD!
# Call:
# VAR(y = x, p = 2, type = "both")

# Estimation results for equation cmort:
# =====
# cmort = cmort.l1 + tempr.l1 + part.l1 + cmort.l2 + tempr.l2 + part.l2 + const + trend
#
#           Estimate Std. Error t value Pr(>|t|)
# cmort.l1  0.297059  0.043734   6.792 3.15e-11 ***
# tempr.l1 -0.199510  0.044274  -4.506 8.23e-06 ***
# part.l1   0.042523  0.024034   1.769  0.07745 .
# cmort.l2  0.276194  0.041938   6.586 1.15e-10 ***
# tempr.l2 -0.079337  0.044679  -1.776  0.07639 .
# part.l2   0.068082  0.025286   2.692  0.00733 **
# const     56.098652 5.916618   9.482 < 2e-16 ***
# trend    -0.011042  0.001992  -5.543 4.84e-08 ***
# ---
# Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
#
#
# Residual standard error: 5.295 on 498 degrees of freedom
# Multiple R-Squared: 0.7227, Adjusted R-squared: 0.7188
# F-statistic: 185.4 on 7 and 498 DF, p-value: < 2.2e-16
```

```

# Estimation results for equation temp:
# =====
# temp = cmort.l1 + temp.l1 + part.l1 + cmort.l2 + temp.l2 + part.l2 + const + trend
#
#           Estimate Std. Error t value Pr(>|t|)
# cmort.l1 -0.108889  0.050667 -2.149  0.03211 *
# temp.l1   0.260963  0.051292  5.088 5.14e-07 ***
# part.l1   -0.050542  0.027844 -1.815  0.07010 .
# cmort.l2 -0.040870  0.048587 -0.841  0.40065
# temp.l2   0.355592  0.051762  6.870 1.93e-11 ***
# part.l2   -0.095114  0.029295 -3.247  0.00125 **
# const     49.880485 6.854540  7.277 1.34e-12 ***
# trend     -0.004754  0.002308 -2.060  0.03993 *
# ---
# Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
#
#
# Residual standard error: 6.134 on 498 degrees of freedom
# Multiple R-Squared: 0.5445, Adjusted R-squared: 0.5381
# F-statistic: 85.04 on 7 and 498 DF, p-value: < 2.2e-16

# Estimation results for equation part:
# =====
# part = cmort.l1 + temp.l1 + part.l1 + cmort.l2 + temp.l2 + part.l2 + const + trend
#
#           Estimate Std. Error t value Pr(>|t|)
# cmort.l1  0.078934  0.091773  0.860 0.390153
# temp.l1   -0.388808  0.092906 -4.185 3.37e-05 ***
# part.l1   0.388814  0.050433  7.709 6.92e-14 ***
# cmort.l2 -0.325112  0.088005 -3.694 0.000245 ***
# temp.l2   0.052780  0.093756  0.563 0.573724
# part.l2   0.382193  0.053062  7.203 2.19e-12 ***
# const    59.586169 12.415669  4.799 2.11e-06 ***
# trend     -0.007582  0.004180 -1.814 0.070328 .
# ---
# Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
#
#
# Residual standard error: 11.11 on 498 degrees of freedom
# Multiple R-Squared: 0.4679, Adjusted R-squared: 0.4604
# F-statistic: 62.57 on 7 and 498 DF, p-value: < 2.2e-16

# Covariance matrix of residuals:
#      cmort temp  part
# cmort 28.034 7.076 16.33
# temp  7.076 37.627 40.88
# part   16.325 40.880 123.45
#
# Correlation matrix of residuals:
#      cmort temp  part
# cmort 1.0000 0.2179 0.2775
# temp  0.2179 1.0000 0.5998
# part   0.2775 0.5998 1.0000

```

By inspecting the top triangle of the correlation matrix of residuals the values are still quite large suggesting not white noise. The not white noise features of the dataset are further exposed in the following `acf` plots of the residuals where lots of values outside of the C.I. intervals.

The Portmanteau test rejects the null hypothesis of white noise, suggesting poor fit:

```
library(vars)
# using fit of p=2 from above code chunk
serial.test(fit, lags.pt = 12, type = "PT.adjusted")
#       Portmanteau Test (adjusted)
#
# data: Residuals of VAR object fit
# Chi-squared = 162.35, df = 90, p-value = 4.602e-06 <- REJECT H_0 SO NOT WHITE NOISE (BAD)!!
#
# This observation is further confirmed by visual inspection:
acf(resid(fit), 52) # lots of acf outside of C.I. so not white noise for upper triangle plots

# Predictions are produced with
fit.pr = predict(fit, n.ahead = 24, ci = 0.95) # 24 weeks ahead
fanchart(fit.pr) # plot prediction + error, graphs show pretty bad predictions
```

9.8 Auto regressive Models with Conditional Heteroscedasticity - ARCH(p) models

- A feature that is frequently observed in asset price data is that a significant change in the price of an asset is often followed by a period of high volatility.
- The class of autoregressive models with conditional heteroskedasticity of order p —the ARCH(p) models—is defined by

$$x_t = \mu + w_t \sqrt{\alpha_0 + \sum_{k=1}^p \alpha_k (x_{t-k} - \mu)^2}, \quad w_t \sim \text{iid } N(0, \sigma_w^2).$$

- Because of the scaling of w_t by past deviations $(x_{t-k} - \mu)^2$, ARCH models capture volatility clustering.
- If $p = 1$, the ARCH(1) model specializes to

$$x_t = \mu + w_t \sqrt{\alpha_0 + \alpha (x_{t-1} - \mu)^2}.$$

- ARCH models are common in financial time series. If z_t is the asset price at day t , then

$$x_t = \log \frac{z_t}{z_{t-1}}$$

is the daily return.

- Note:
 - Setting $\alpha_k = 0$ for all $k \geq 1$ “switches off” the ARCH effect, reverting to white noise of variance α_0 .
 - We are modelling only the conditional variance of x_t given its past p values.
 - Extending to an ARMA mean structure yields GARCH models.

10 Module 10 - Estimation and Forecasting

10.1 Behaviour of the ACF and PACF for ARMA Models

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

- The PACF for MA models behaves much like the ACF for AR models.
- The PACF for AR models behaves much like the ACF for MA models.
- Since an invertible ARMA model has an infinite AR representation, its PACF will not cut off.
- Since a causal ARMA model has an infinite MA representation, its ACF will not cut off.
- Remember that the data might have to be detrended and/or transformed first (e.g., to stabilise the variance, apply a log transform), before such analysis is performed.

10.2 Behaviour of ACF and PACF in presence of seasonality

	AR($P)_s$	MA($Q)_s$	ARMA($P, Q)_s$
ACF*	Tails off at lags ks , $k = 1, 2, \dots$	Cuts off after lag Qs	Tails off at lags ks , $k = 1, 2, \dots$
PACF*	Cuts off after lag Ps	Tails off at lags ks , $k = 1, 2, \dots$	Tails off at lags ks , $k = 1, 2, \dots$

*The values at nonseasonal lags $h \neq ks$, for $k = 1, 2, \dots$, are zero.

- This can be seen as a generalisation of the previous table (which is the special case $s = 1$).
- Fitting seasonal auto regressive and moving average components **first** generally leads to more satisfactory results.

10.3 Box-Jenkins Methodology

- Determine the integration order d (of the $ARIMA(p, d, q)$) and work on the corresponding $ARMA(p, q)$ by differencing.
- Then choose candidates for p and q from examining ACF and PACF
- For fixed p and q (candidates), estimate parameters via:
 - Method of moments
 - Maximum likelihood (much less efficient when $q > 0$)
 - Least squares and variations
- Perform diagnostics, to choose the best p and q . Thus may suggest new candidates for p and q , in which case we perform a new iteration from step 2.
- Use the chosen model for forecasting.

Example of Box-Jenkins steps:

Step 1 (choosing d):

Choosing d involves:

- A time series x_t may be modelled by a stationary ARMA model if the sample ACF decays rapidly. If the decay is slow (and the ACF is positive) then this suggests further differencing.
- Let σ_d^2 be the sample variance of the $\nabla^d x_t$. This quantity should first decrease with d until stationarity is achieved, and then start to increase. Hence d could be set to the value which minimises σ_d^2 , $d \geq 0$.

Be careful not to over-difference, as this introduces artificial dependence in the data.

FYI WITH SEASONALITY IF $d + D \leq 1$ initially fit a constant term then drop if it is not significant.

Step 2 (choosing p and q):

- We work on the differenced series, which we assume is $ARMA(p, q)$ and has mean 0. If the mean is $\neq 0$, subtract its sample mean, and work on the residuals.
- Examine ACF and PACF to choose candidates for p and q :
 - p can be inferred from the number of spikes in the PACF until a geometrical decay to zero is observed.
 - q can be inferred from the number of spikes in the ACF until a geometrical decay to zero is observed.
- alternatively, work iteratively from the simplest models and increase orders p and q :
 - Higher orders will always reduce the sum of squares (more parameters) - at the extreme model with n parameters will replicate the data.
 - The approximate order could be chosen by using information criteria (e.g. BIC or AIC).

```
acf2(rec, 48) # recruitment series for 48 months of lag
```

The ACF decays straight away (so $q \approx 0$) and PACF has 2 spikes then decays (so $p \approx 2$).

Step 3 (estimation of parameters):

- We now need estimates for ϕ_1, \dots, ϕ_p and $\theta_1, \dots, \theta_q$ for given p and q
- Use R to fit model with `fit <- arima(x, order = c(p, 0, q))`, where x is differenced series and where p and q are the values from before.
- The default estimation procedure (unless there are missing values) is to use conditional sum-of-squares to find starting values, then maximum likelihood.
- This output parameter estimates, their standard errors, as well as σ^2 and the AIC.
- One could alternatively use methods of moments approach (see later), but this is much less efficient than MLE if $q > 0$.

Code for MLE values for just AR general case below

```
rec.mle = ar.mle(rec, order = 2)
rec.mle$x.mean
# [1] 62.26153
rec.mle$ar
# [1] 1.3512809 -0.4612736
sqrt(diag(rec.mle$asy.var.coef))
# [1] 0.04099159 0.04099159
rec.mle$var.pred # the PROPORTION NOT EXPLAINED by the model (so not good)
# [1] 89.33597

# The fit is displayed as follows
ts.plot(rec, main = "Results of fit using the R MLE estimators")
lines(rec[1:length(rec)] - rec.mle$resid, col = "blue", lwd = 1)
```

Best estimators general ARMA case (gives very similar values to above, but uses different estimation procedure):

```
rec.arima0 <- arima(rec, order = c(2, 0, 0)) # to use with tsdiag

rec.arima0$coef[3]
# intercept
# 61.85847
rec.arima0$coef[1:2]
#      ar1      ar2
# 1.3512196 -0.4612128
sqrt(diag(rec.arima0$var.coef))
#      ar1      ar2  intercept
# 0.04158461 0.04166812 4.00393366
rec.arima0$sigma2
# [1] 89.33436

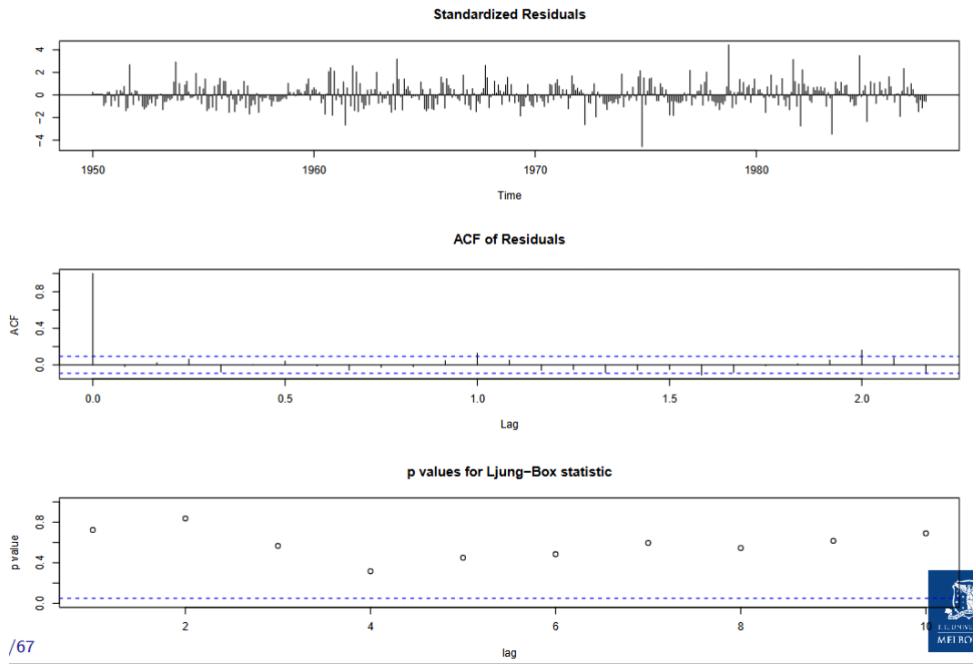
# The fit is displayed as follows
ts.plot(rec, main = "Residuals of fit using R ARIMA estimators")
lines(rec[1:length(rec)] - rec.arima0$residuals, col = "blue", lwd = 1)
```

Step 4 (diagnostics):

Guiding principle: if we have a good fit, the residuals should be (uncorrelated) white noise. This can be done in three ways (can be done in R using `tsdiag(fit)`, but better to use `sarima(x, p, d, q from astsa)`):

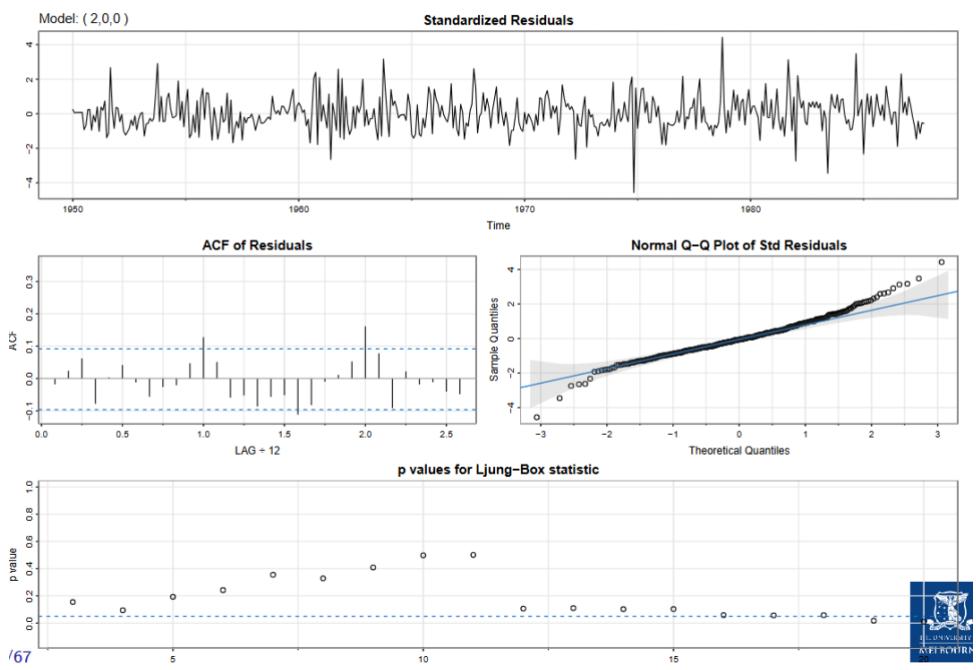
1. **Residuals:** If visual inspection of the residuals exhibit any pattern (trend or magnitude) then these are not white noise.
2. **ACF and PACF:** These should be within the confidence intervals with approximate frequency (95%)
3. **Ljung-Box Portmanteau Test** (most formal): This test considers the dimensions of the models (number of parameters), and tests whether correlations are 0 at all lags, and displays p -values. High p -values mean we cannot reject the (null) hypothesis of white noise (which is what we want).

```
tsdiag(rec.arima0)
```



Notes: Too many spikes (3/26) outside of confidence interval, so bad.

```
RecSARIMAdig <- sarima(rec, 2, 0, 0)
```



Notes: In Q-Q plot lots of dots outside of C.I. (bad), too many ACF spikes outside of C.I. (bad), and sharp drop in p -values for Ljung-Box statistic (bad, sharp drop at 12 suggests seasonality).

```
RecSARIMAdig$fit
```

```
# Call:  
# arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q),  
#       xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed =  
#       optim.control = list(trace = trc, REPORT = 1, reltol = tol))  
#
```

```

# Coefficients:
#      ar1     ar2   xmean
#      1.3512 -0.4612 61.8585
# s.e.  0.0416  0.0417  4.0039
#
# sigma^2 estimated as 89.33: log likelihood = -1661.51, aic = 3331.02 <- AIC FOR WHOLE MODEL
RecSARIMADiag$ICs
#      AIC     AICc      BIC
# 7.353244 7.353362 7.389587 <- GIVES AIC PER DATA POINT (to get above do 7.35 x # data points)

```

10.3.1 Example: Air Passengers

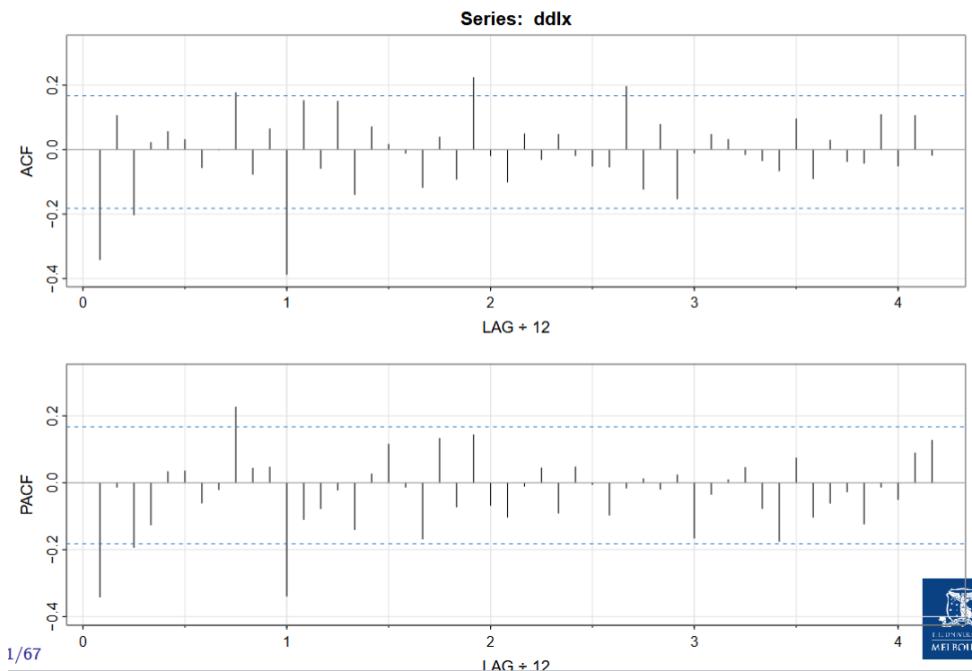
From before we already found that taking 12th order difference (seasonal), on the differenced logged data makes the residuals look reasonably stationary.

```

x = AirPassengers
lx = log(x)
dlx = diff(lx)
ddlx = diff(dlx, 12)

acf2(ddlx, 50) # ACF and PACF for data

```



Seasonal component:

- At the seasons, the ACF appears to be cutting off at lag 1_s ($s = 12$)
- PACF appears to be tailing off at lags 1_s, 2_s, 3_s, 4_s, ...
- $\Rightarrow SMA(1)$, $P = 0$, $Q = 1$, in the seasons $s = 12$

Non-seasonal component:

- We inspect the ACF and PACF at lower lags
- Both appear to be tailing off, which suggests ARMA within the seasons, say with $p = q = 1$
- $\Rightarrow ARMA(1, 1)$

Thus, we will first try an $ARIMA(1, 1, 1) \times (0, 1, 1)_{12}$ on the logged data

```

AirPassFit1 = sarima(lx, 1, 1, 1, 0, 1, 1, 12)
# sarima(data, p, d, q, P, D, Q, s)

```

```

# could use dd1x: sarima(dd1x, 1, 0, 1, 0, 0, 1, 12)
AirPassFit1$fit
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q),
## include.mean = !no.constant, transform.pars = trans, fixed = fixed,
## REPORT = 1, reltol = tol))
##
## Coefficients:
##       ar1     ma1    sma1
##      0.1960 -0.5784 -0.5643
## s.e. 0.2475  0.2132  0.0747 # <- AR parameter is not significant too high, so drop it
##
## sigma^2 estimated as 0.001341: log likelihood = 244.95, aic = -481.9

# We can try other models to test
AirPassFit2 <- sarima(lx, 0, 1, 1, 0, 1, 1, 12) # ARIMA(0,1,1)x(0,1,1)_12
AirPassFit3 <- sarima(lx, 1, 1, 0, 0, 1, 1, 12) # ARIMA(1,1,0)x(0,1,1)_12

AirPassFit2$fit
## Coefficients:
##       ma1    sma1
##      -0.4018 -0.5569
## s.e. 0.0896 0.0731
##
## sigma^2 estimated as 0.001348: log likelihood = 244.7, aic = -483.4

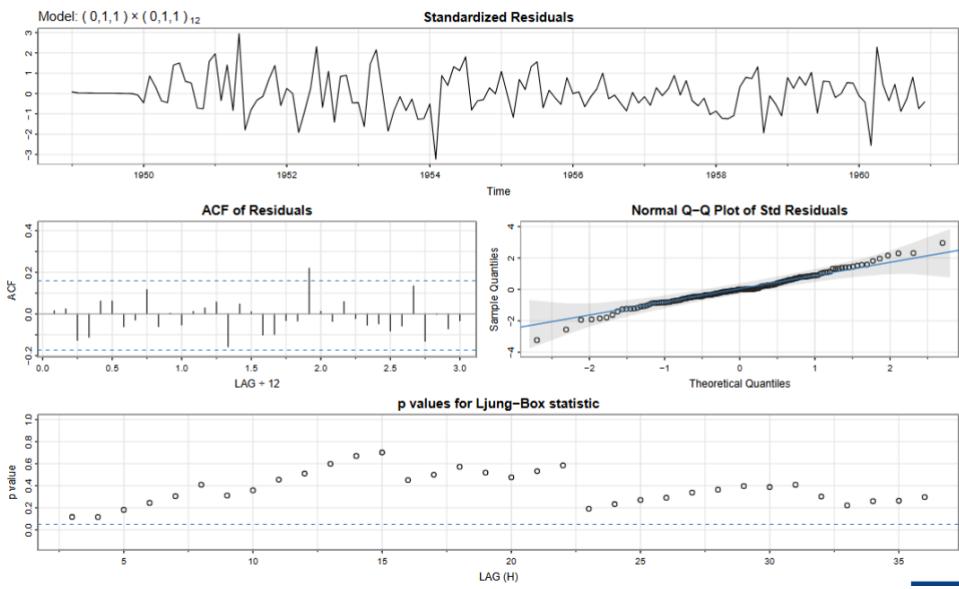
AirPassFit3$fit
## Coefficients:
##       ar1     sma1
##      -0.3395 -0.5619
## s.e. 0.0822 0.0748
##
## sigma^2 estimated as 0.001367: log likelihood = 243.74, aic = -481.49

AirPassFit2$ICs
##      AIC      AICC      BIC
## -3.690069 -3.689354 -3.624225

AirPassFit3$ICs
##      AIC      AICC      BIC
## -3.675493 -3.674777 -3.609649

# INFORMATION CRITERIA FOR Fit2 is LOWER SO BETTER, SO TRY THAT MODEL!
AirPassFit2 <- sarima(lx, 0, 1, 1, 0, 1, 1, 12) # ARIMA(0,1,1)x(0,1,1)_12

```



Except for 1-2 outliers, the model fits well.

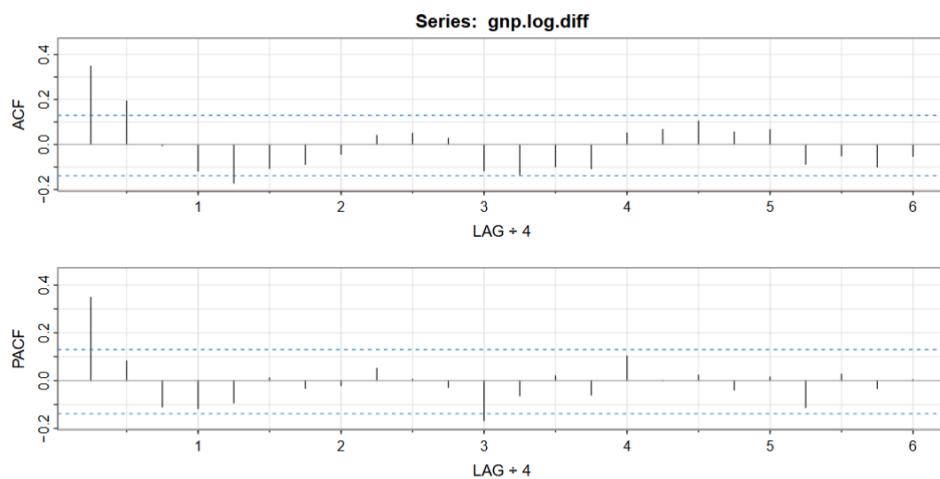
10.3.2 Example: GNP data

```
plot(gnp, main = "Quartely U.S. GNP")
# data seems to have exponential growth, so a log transformation is appropriate

acf2(gnp, 50, main = "Sample ACF and PACF of the GNP data")
# slow decay of ACF => differencing may be appropriate

# Making the modifications:
gnp.log.diff = diff(log(gnp)) # growth rate plot(gnpr)
ts.plot(gnp.log.diff, main="")
abline(mean(gnp.log.diff), 0, col = "blue", lwd = 2) # plots mean

acf2(gnp.log.diff, 24)
```



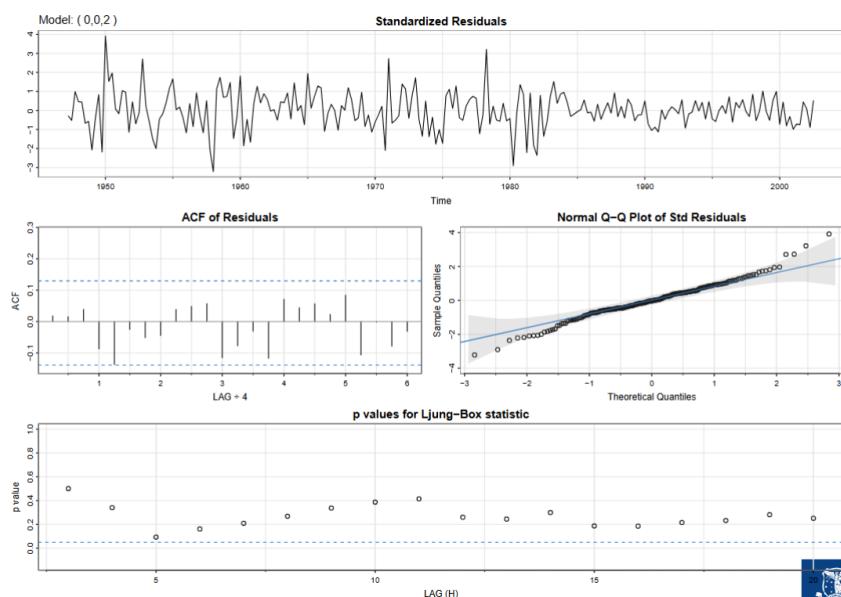
Note: we can see the ACF cuts off after second lag (potential $MA(2)$). The PACF seems to cut off after first lag (potential $AR(1)$)

GNP data: $MA(2)$ fit

```

GNP.MA2 <- sarima(gnp.log.diff, 0, 0, 2) # MA(2)
GNP.MA2$fit
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed =
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##      ma1    ma2   xmean
##      0.3028 0.2035 0.0083
##  s.e. 0.0654 0.0644 0.0010
## 
## sigma^2 estimated as 8.919e-05: log likelihood = 719.96, aic = -1431.9
GNP.MA2$ICs
##      AIC      AICc      BIC
## -6.450133 -6.449637 -6.388823

```



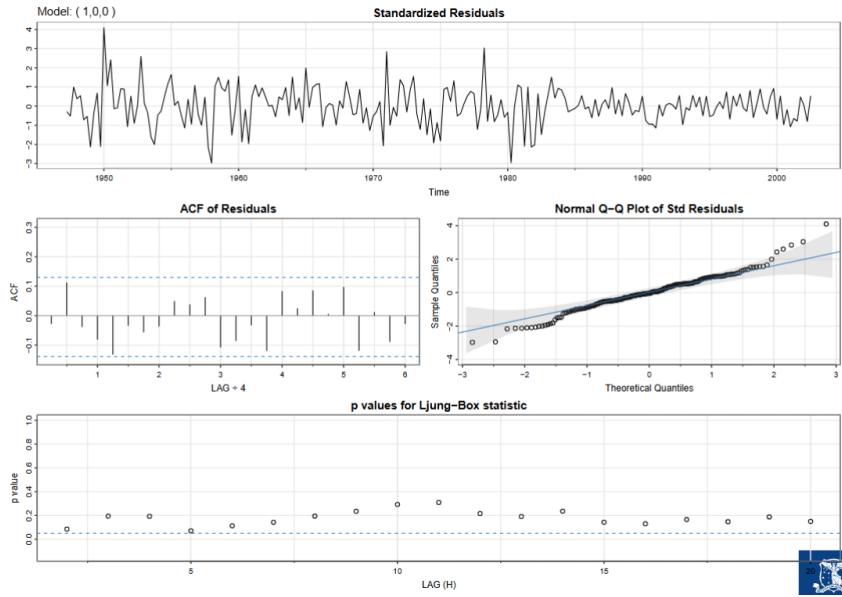
GNP data: AR(1) fit

```

GNP.AR1 <- sarima(gnp.log.diff, 1, 0, 0) # AR(1)
GNP.AR1$fit

## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed =
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##      ar1   xmean
##      0.3467 0.0083
##  s.e. 0.0627 0.0010
## 
## sigma^2 estimated as 9.03e-05: log likelihood = 718.61, aic = -1431.22
GNP.AR1$ICs
##      AIC      AICc      BIC
## -6.446940 -6.446693 -6.400958

```

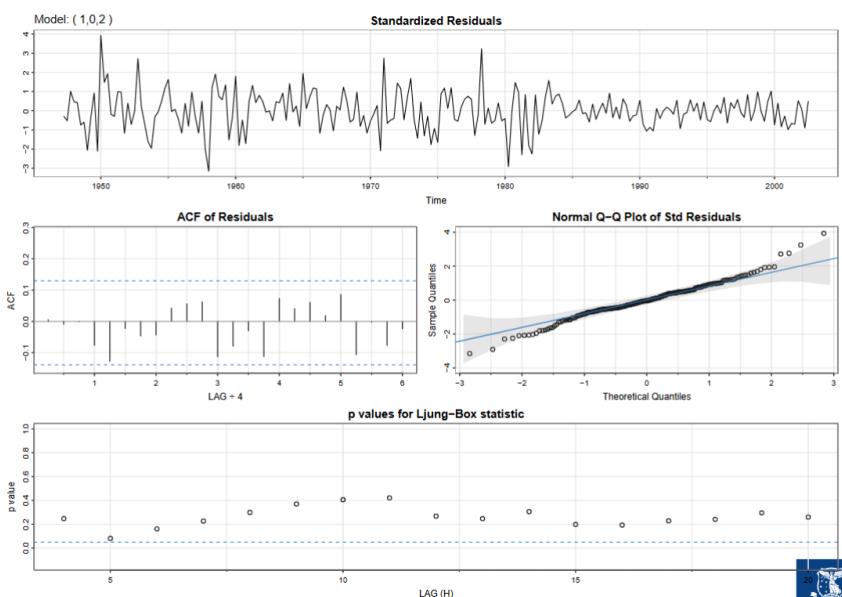


GNP data: ARMA(1,2) fit

```

GNP.ARMA <- sarima(gnp.log.diff, 1, 0, 2) # ARMA(1,2)
GNP.ARMA$fit
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed =
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##      ar1    ma1    ma2   xmean
##      0.2407 0.0761 0.1623 0.0083
## s.e. 0.2066 0.2026 0.0851 0.0010 <- SE VERY HIGH for AR1, MA1!!!!
##
## sigma^2 estimated as 8.877e-05: log likelihood = 720.47, aic = -1430.9
GNP.ARMA$ICs
##      AIC      AICc      BIC
## -6.445712 -6.444882 -6.369075

```



But the graphs look pretty good!

GNP data: Comparison

Note the fitted models are

- $MA(2) : \hat{x}_t = 0.008_{(0.001)} + 0.303_{(0.065)}\hat{w}_{t-1} + 0.204_{(0.064)}\hat{w}_{t-2} + \hat{w}_t$, where $\hat{\sigma}_w = 0.0094$.
- $AR(1) : \hat{x}_t = 0.008_{(0.001)}(1 - 0.347) + 0.347_{(0.0063)}\hat{x}_{t-1} + \hat{w}_t$, where $\hat{\sigma}_w = 0.0095$.
- Both models are nearly the same. This is because the AR model can be rewritten (ignoring the constant) as: $x_t \approx 0.35w_{t-1} + 0.12w_{t-2} + w_t$, where the coefficients are obtained by

```
formatC(ARMAtoMA(ar=0.35, ma=0, 6), digits=3)
## [1] "0.35" "0.122" "0.0429" "0.015" "0.00525" "0.00184"
```

GNP data: Model Selection

- Information criteria (the lower the better):
 - AR(1): \$AIC: -6.446940 \$AICc: -6.446693 \$BIC: -6.400958
 - MA(2): \$AIC: -6.450133 \$AICc: -6.449637 \$BIC: -6.388823
 - ARMA(1,2): \$AIC: -6.445712 \$AICc: -6.444882 \$BIC: -6.369075
- The AIC and AICc both prefer the $MA(2)$ fit to $AR(1)$
- The BIC prefers the simpler $AR(1)$ model to $MA(2)$
- It is often the case that the BIC will select a model of smaller order than the AIC or AICc. In either case, it is not unreasonable to retain the $AR(1)$ because pure AR models are easier to work with.
- Combining the two to $ARMA(1, 2)$ leads to poorer scores.

10.4 Forecasting - Basics

10.4.1 Best Linear Predictors and Prediction for Stationary Processes

Here we restrict our attention to predictors that are linear functions of the data, that is predictors of the form:

$$x_{n+m}^n = \alpha_0 + \sum_{k=1}^n \alpha_k x_k,$$

where $\alpha_0, \alpha_1, \dots, \alpha_n$ are real numbers and are the best linear predictors. Note:

- The α 's depend on m too, but that is not reflected in the notation.
- Such estimators depend only on the second-order moments of the process, which are easy to estimate from the data.

BLP for Stationary Processes

Given data x_1, \dots, x_n , the best linear predictor

$$x_{n+m}^n = \alpha_0 + \sum_{k=1}^n \alpha_k x_k$$

of x_{n+m} for $m \geq 1$ is found by solving

$$\mathbb{E}[(x_{n+m} - x_{n+m}^n)x_k] = 0, \quad k = 0, 1, \dots, n$$

where $x_0 = 1$, for $\alpha_0, \alpha_1, \dots, \alpha_n$

- The $n + 1$ equations specified above are called the **prediction equations**, and they are used to solve for the coefficients $\{\alpha_0, \alpha_1, \dots, \alpha_n\}$
- This result stems from minimising least squares.

If $\mathbb{E}[x_t] = \mu$, the first equation ($k = 0$) implies

$$\mathbb{E}[x_{n+m}^n] = \mathbb{E}[x_{n+m}] = \mu$$

Thus, taking expectation of the BLP leads to

$$\mu = \alpha_0 + \sum_{k=1}^n \alpha_k \mu \text{ or } \alpha_0 = \mu \left(1 - \sum_{k=1}^n \alpha_k \right)$$

Hence, the form of the BLP is

$$x_{n+m}^n = \mu + \sum_{k=1}^n \alpha_k (x_k - \mu_k)$$

Henceforth, we will assume that $\mu = 0 \Leftrightarrow \alpha_0 = 0$, wlog, as long as parameters are assumed known.

10.4.2 One-step-ahead prediction

- Given $\{x_1, x_2, \dots, x_n\}$ we wish to forecast the time series x_{n+1} value at the next point in time $n + 1$.
- The BLP of x_{n+1} is of the form

$$x_{n+1}^n = \phi_{n1} x_n + \phi_{n2} x_{n-1} + \dots + \phi_{nn} x_1$$

- In this case, α_k is $\phi_{n,n+1-k}$ for $k = 1, \dots, n$.
- Using BLP result above, the coefficients of ϕ_{ni} 's satisfy

$$\sum_{j=1}^n \phi_{nj} \gamma(k-j) = \gamma(k) \text{ for } k = 1, \dots, n$$

- Or can be written in matrix form $\Gamma_n \phi_n = \gamma_n$
- Where $\Gamma_n = \{\gamma(k-j)\}_{j,k=1}^n$, $\phi_n = (\phi_{n1}, \dots, \phi_{nn})'$. Also $\gamma_n = (\gamma(1), \dots, \gamma(n))'$
- The matrix Γ_n is non-negative definite (in fact positive definite for ARMA models).

We have then

$$\phi_n = \Gamma_n^{-1} \gamma_n$$

- The one-step-ahead forecast is then

$$x_{n+1}^n = \phi_n' x, \text{ where } x = (x_n, x_{n-1}, \dots, x_1)'$$

- The mean square one-step-ahead prediction error is

$$P_{n+1}^n = \mathbb{E}[(x_{n+1} - x_{n+1}^n)^2] = \gamma(0) - \gamma_n' \Gamma_n^{-1} \gamma_n$$

10.5 Forecasting

We assume that x_t is a causal and invertible $ARMA(p, q)$ process

$$\phi(B)x_t = \theta(B)w_t, \text{ where } w_t \sim \text{iid } N(0, \sigma_w^2)$$

Note: if there is non-zero mean $\mathbb{E}[x_t] = \mu_x$, simple replace x_t with $x_t - \mu_x$ in the model.

Two types of forecasts:

1. The minimum mean square error predictor of x_{n+m} , defined as

$$x_{n+m}^n = \mathbb{E}[x_{n+m}|x_n, x_{n-1}, \dots, x_1]$$

2. The predictor of x_{n+m} based on the *infinite* past, defines as

$$\tilde{x}_{n+m} = \mathbb{E}[x_{n+m}|x_n, x_{n-1}, \dots, x_1, x_0, x_{-1}, \dots]$$

Note:

- For ARMA models, \tilde{x}_{n+m} is easier to calculate.
- In general, x_{n+m}^n and \tilde{x}_{n+m} are not the same.
- For large samples, \tilde{x}_{n+m} provides a good approximation to x_{n+m}^n .

10.5.1 Forecasts for more than one step

Write x_{n+m} in its invertible form:

$$w_{n+m} = \sum_{j=0}^{\infty} \pi_j x_{n+m-j}, \quad \pi_0 = 1$$

Taking conditional expectations we have

$$0 = \tilde{x}_{n+m} + \sum_{j=1}^{\infty} \pi_j \tilde{x}_{n+m-j} \quad \text{since } \pi_0 = 1$$

Because $\mathbb{E}[x_t|x_n, x_{n-1}, \dots, x_0, x_{-1}, \dots] = x_t$ for $t \leq n$ this can be rewritten as

$$\tilde{x}_{n+m} = - \sum_{j=1}^{m-1} \pi_j \tilde{x}_{n+m-j} - \sum_{j=m}^{\infty} \pi_j x_{n+m-j}$$

Prediction is then accomplished recursively using this formula, start with the one-step-ahead ($m = 1$), and then continuing for $m = 2, 3, \dots$

Finding mean-square prediction error:

Write x_{n+m} in its causal form:

$$x_{n+m} = \sum_{j=0}^{\infty} \psi_j w_{n+m-j}, \quad \psi_0 = 1$$

Taking conditional expectations we have

$$\tilde{x}_{n+m} = \sum_{j=0}^{\infty} \psi_j \tilde{w}_{n+m-j} = \sum_{j=m}^{\infty} \psi_j w_{n+m-j}$$

Because

$$\tilde{w}_t = \mathbb{E}[w_t | x_n, x_{n-1}, \dots, x_0, x_{-1}, \dots] = \begin{cases} 0 & t > n, \\ w_t & t \leq n. \end{cases}$$

Mean-square prediction error can now be calculated since we now know that

$$\tilde{x}_{n+m} = \sum_{j=m}^{\infty} \psi_j w_{n+m-j} \implies x_{n+m} - \tilde{x}_{n+m} = \sum_{j=0}^{m-1} \psi_j w_{n+m-j}$$

and hence

$$P_{n+m}^n = \mathbb{E}[(x_{n+m} - \tilde{x}_{n+m})^2] = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2$$

Note that for a fixed sample size, n , the prediction errors are correlated. That is, for lag $k \geq 1$,

$$\mathbb{E}[(x_{n+m} - \tilde{x}_{n+m})(x_{n+m+k} - \tilde{x}_{n+m+k})] = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j \psi_{j+k}$$

10.5.2 Predictions in practice

- When n is small the system of "prediction equations" (where $x_0 = 1$)

$$\mathbb{E}[(x_{n+m} - \tilde{x}_{n+m})x_k] = 0, \quad k = 0, 1, \dots, n$$

- However, when n is large one will want to use, but in practice we do not observe $x_0, x_{-1}, x_{-2}, \dots$ and only x_1, x_2, \dots, x_n are available. We then need to truncate the infinite sum on the RHS

$$\tilde{x}_{n+m} = - \sum_{j=1}^{m-1} \pi_j \tilde{x}_{n+m-j} - \sum_{j=m}^{\infty} \pi_j x_{n+m-j}$$

- The **truncated predictor** is then written as (which is applied recursively)

$$\tilde{x}_{n+m}^n = - \sum_{j=1}^{m-1} \pi_j \tilde{x}_{n+m-j}^n - \sum_{j=m}^{n+m-1} \pi_j x_{n+m-j}$$

- The mean squared error, in this case, is approximated using P_{n+m}^n as before.

10.5.3 Truncated prediction for ARMA

For $ARMA(p, q)$ models, the truncated predictors for $m = 1, 2, \dots$ are

$$\tilde{x}_{n+m}^n = \phi_1 \tilde{x}_{n+m}^n + \dots + \phi_p \tilde{x}_{n+m-p}^n + \theta_1 \tilde{w}_{n+m-1}^n + \dots + \theta_q \tilde{w}_{n+m-q}^n$$

, where $\tilde{x}_t^n = x_t$ for $1 \leq t \leq n$ and $\tilde{x}_t^n = 0$ for $t \leq 0$. The truncated prediction errors are given by

$$\tilde{w}_t^n = \begin{cases} 0 & t \leq 0 \text{ or } t > n \\ \phi(B) \tilde{x}_n^n - \theta_1 \tilde{w}_{t-1}^n - \dots - \theta_q \tilde{w}_{t-q}^n & 1 \leq t \leq n \end{cases}$$

Note:

- For $AR(p)$ models with $n > p$ (so there is no need for approximations),

$$\tilde{x}_{n+m}^n = \tilde{x}_{n+m} = x_{n+m}^n$$

- The above approximation is required for $MA(q)$ and $ARMA(p, q)$ where $q > 0$.

10.5.4 Long-range forecasts

- Consider forecasting an ARMA process with mean μ_x
- Replacing x_{n+m} with $x_{n+m} - \mu_x$ in the causal representation above, and taking expectation in an analogous way implies that the m -step-ahead forecast can be written as

$$\tilde{x}_{n+m} = \mu_x + \sum_{j=m}^{\infty} \psi_j w_{n+m-j}$$

- Because the ψ dampen to zero exponentially fast,

$$\tilde{x}_{n+m} \rightarrow \mu_x \text{ exponentially fast as } m \rightarrow \infty$$

- Moreover, the mean squared prediction error

$$P_{n+m}^n \rightarrow \sigma_w^2 \sum_{j=0}^{\infty} \psi_j^2 = \gamma_x(0) = \sigma_x^2 \text{ exponentially fast as } m \rightarrow \infty$$

- ARMA forecasts quickly settle to the mean with a constant prediction error as forecast horizon, m , grows.

Forecasting example on recruitment data

```
rec.arima0 <- arima(rec, order = c(2, 0, 0))
fore2 <- predict(rec.arima0, n.ahead=36)
cbind(fore2$pred, fore2$se)
##          fore2$pred fore2$se
## Oct 1987 20.36547 9.451686
## Nov 1987 26.08036 15.888378
## Dec 1987 32.65148 20.464325
## Jan 1988 38.89474 23.492457
## Feb 1988 44.30006 25.393693
## Mar 1988 48.72437 26.537088
## Apr 1988 52.20958 27.199368
## May 1988 54.87831 27.570234
## Jun 1988 56.87693 27.771616
## Jul 1988 58.34666 27.877923
## Aug 1988 59.41079 27.932597
## Sep 1988 60.17081 27.960042
## Oct 1988 60.70697 27.973508
## Nov 1988 61.08091 27.979974
## Dec 1988 61.33890 27.983014

ts.plot(rec, fore2$pred, col=1:2, xlim=c(1980, 1990.5), ylab="Recruitment")
U = fore2$pred + fore2$se
L = fore2$pred - fore2$se
xx = c(time(U), rev(time(U)))
yy = c(L, rev(U))
polygon(xx, yy, border = 8, col = gray(0.6, alpha=0.2)) # shows CI
lines(fore2$pred, type="p", col=2) # makes red points for forecasts, converges to mean as expected
```

Forecasting example on Air Passenger data

```
sarima.fore(1x, 12, 0, 1, 1, 0, 1, 1, 12) # the second param gives # forecasts (=12)
```

11 Appendix A - Midsemester Revision Sheet

refer to next page

M2 - Collective Risk Modelling

Discrete Convolution: $F_{X+Y}(x) = \sum_y F_Y(y) f_X(x)$, $f_{X+Y}(x) = \sum_y f_Y(y) f_X(x)$

Continuous Convolution: $F_{X+Y}(x) = \int_{-\infty}^x F_Y(s-x) f_X(x) dx$, $f_{X+Y}(x) = \int_{-\infty}^x f_Y(s-x) f_X(x) dx$

Collective Risk Model: $S = \sum_{i=1}^N Y_i$, $E[S] = E[N]E[Y]$, $\text{Var}(S) = E[N]E[Y^2] + [E[Y]^2](\text{Var}(N) - E[N])$, $M_N(\log(M_Y(t)))$. $F_S(x) = \sum_{n=0}^{\infty} P(S \leq x | N=n) P(N=n)$

Always discrete mass at 0, discrete/continuous/mixed for > 0 depending on Y

if Y is discrete: $f_S(s) = \sum_{n=0}^{\infty} P(S=s | N=n) P(N=n)$

Binomial Distribution: $N \sim \text{Bin}(v, p)$, $P(N=k) = \binom{v}{k} p^k (1-p)^{v-k}$, $\binom{v}{k} = \frac{v!}{k!(v-k)!}$

Compound Binomial Distribution $S \sim \text{CompBinom}(v, p, G)$, also if S_1, \dots, S_n :

$\sum_{i=1}^n S_i \stackrel{d}{=} \text{CompBinom}\left(\sum_{i=1}^n v_i, p, G\right)$ * G is individual claim size distribution*
e.g. $S_{2c} = \sum_{i=1}^n Y_i \mathbb{1}_{\{Y_i > M\}} \stackrel{d}{=} \text{CompBinom}(v, p(1-G(M)), G_{2c})$, where $G_{2c} = P(Y_i \leq y | Y_i > M)$

Poisson Distribution: $N \sim \text{Pois}(\lambda v)$, $P(N=n) = \frac{e^{-\lambda v} (\lambda v)^n}{n!}$ (as #trials $\rightarrow \infty$: binomial = poisson)

Compound Poisson Distribution $S \sim \text{CompPois}(\lambda v, G)$

Overdispersed: Variance is larger than mean (implemented in mixed poisson)

Mixed Poisson Distribution: Assume $\Lambda \sim H$ ($H(0)=0$, $E[\Lambda]=\lambda$, $\text{Var}(\Lambda)>0$), $N| \Lambda \sim \text{Pois}(\lambda v)$

$P(N=n) = \int_0^\infty \frac{e^{-\lambda v} (\lambda v)^n}{n!} \cdot h(\lambda) d\lambda$, $E[N] = E[\Lambda]v = \lambda v$, $\text{Var}(N) = \lambda v + v^2 \text{Var}(\Lambda) > E[N]$,

$M_N(t) = E[E[e^{tN} | \Lambda]] = E[e^{\lambda v(e^{t-1})}] = M_\Lambda(v(e^{t-1}))$

Negative Binomial Distribution ($\Lambda = \lambda \oplus, \oplus \sim T(r, \gamma)$): $\sigma_{\oplus} = \frac{1}{r}$ $\sigma_{\Lambda}^2 = \lambda^2/r$

$N| \oplus \stackrel{d}{=} \text{NegBin}(\lambda v, r)$, proof with mgf of N , $p = \frac{\lambda v}{\lambda v + r}$, $r^{\text{th}} \text{ success}$: $P_R = \binom{k+r-1}{k} p^k (1-p)^r$

\Rightarrow Interpretation: $E[N] = \lambda v$, $V(N) = \lambda v(1 + \frac{\lambda v}{r}) > E[N]$, $V_{co}(\frac{N}{v}) = \frac{v}{r} = \frac{1}{\lambda v} + \frac{1}{r}$

\Rightarrow Additional uncertainty not diversifiable $V_{co}(\frac{N}{v}) \rightarrow \frac{1}{r} > 0$ as $v \rightarrow \infty$

Compound Negative Binomial Distribution: $S \sim \text{CompNB}(\lambda v, r, G)$. $N \sim \text{NegBin}(\lambda v, r)$

Aggregation Property: let $S_i \sim \text{CompPoi}(\lambda_i v_i, G_i) \Rightarrow S = \sum_{i=1}^n S_i$, $v_i = \sum_j v_{ij}$, $\lambda_i = \sum_j \lambda_{ij}$, $G = \sum_i \frac{\lambda_i v_i}{\lambda v} G_i$

e.g. $S_i = Y_i N_i$ ($v_i=1$), y_i is degen. dist. ($G(y_i)=1$, $G(y)=0$, $y_i \neq y$), $S \sim \sum_i S_i \sim \text{CompPois}(\lambda v, G)$, $Y = \min(S, d)$, $Z = \max(S-d, 0)$

$v = \sum_i v_i = m$, $\lambda = \sum_i \frac{v_i}{v} \lambda_i = \frac{\sum_i \lambda_i v_i}{m}$, $G(y) = \sum_i \frac{\lambda_i v_i}{\lambda v} G_i(y_i) = \frac{\lambda_i}{\sum_i \lambda_i}$

in each Partition A_k : $Y_i = Y_i \mathbb{1}_{\{Y_i \in A_k\}} + \dots + Y_i \mathbb{1}_{\{Y_i \in A_m\}}$, $v_k = v$, $\lambda_k = \lambda p^{(k)}$

\Rightarrow meaning the volume remains constant in each partition, but the claims frequencies λ_k change proportionally to the probabilities of falling in each partition A_k (thinning of poiss process)

$\Rightarrow S_k = \sum_{i=1}^n Y_i \mathbb{1}_{\{(Y_i, I_i) \in A_k\}} \sim \text{CompPois}(\lambda_k v_k, G_k)$, $\lambda_k v_k = \lambda v p^{(k)} > 0$, $G_k(y) = P(Y_i \leq y | (Y_i, I_i) \in A_k)$

Sparse Vector Algorithm: $S \sim \text{CompPois}(\lambda, g(y_i) = \gamma_i)$, $i=1, \dots, m$. $S = y_1 N_1 + \dots + y_m N_m$

N_1, \dots, N_m , $N_i \sim \text{Poi}(\lambda = \lambda \pi_i)$, $\pi_i(x) = P(iN_i=x) = P(N_i = \frac{x}{\pi_i})$

Large Claim Separation: let M be claim threshold. $S_{sc} = \sum_{i=1}^n Y_i \mathbb{1}_{\{Y_i \leq M\}}$, $S = S_{sc} + S_{nc}$, $S_{sc} \sim \text{CompPois}(\lambda_{sc} v, G_{sc}(y) = P(Y_i \leq y | Y_i \leq M))$

$\lambda_{sc} v = \lambda (1 - G(M)) v$, $G_{sc}(y) = P(Y_i \leq y | Y_i > M)$, proof using mgf

e.g. $M S_{sc}(r) = E[e^{r \sum_{i=1}^n Y_i \mathbb{1}_{\{Y_i \leq M\}}}] = E\left[\prod_{i=1}^n e^{r Y_i \mathbb{1}_{\{Y_i \leq M\}}}\right] = E\left[\prod_{i=1}^n (e^{r Y_i} \mathbb{1}_{\{Y_i \leq M\}} + 1 - e^{r Y_i} \mathbb{1}_{\{Y_i > M\}})\right]$

$= E[e^{r Y_i} \mathbb{1}_{\{Y_i \leq M\}}] [1 - G(M)] + G(M) = M_N(\log(M_{Y_i \mid Y_i > M}(r) (1 - G(M)) + G(M)))$

M3 - Individual Claim Size Modelling

Zero Inflated Severity Model ($X=IB$): $P(I=1)=q$, $P(I=0)=1-q$, $F_X(x) = 1-q + q F_B(x)$, $M_X(t) = 1-q + q M_B(t)$, $E[X] = q E[B]$, $V(X) = q(1-q) E[B]^2 + q V(B)$, $E[X^2] = E[E[B^2], I^2 V(B)]$
case $B=b$: $E[X] = bq$, $V(X) = b^2 V(I) = b^2 q(1-q)$

Likelihood (with left truncation/right censoring): (t_i, x_i, δ_i) , t_i = left truncation point, x_i = claim value, δ_i = indicator if limit has been reached, likelihood:

$L(\theta; \vec{x}) = \prod_i \left[\frac{f(x_i; \theta)}{1 - F(t_i; \theta)} \right]^{1-\delta_i} \frac{F(t_i; \theta)^{\delta_i}}{1 - F(x_i; \theta)}$, $\ell = \max \text{ log-likelihood}$

Akaike Information Criteria: $AIC = -2 \ell + 2d$, $d = \# \text{parameters estimated}$

Bayesian Information Criteria: $BIC = -2 \ell + \log(n)d$, $n = \# \text{parameters}$

Deductibles: $Y = \max(0, X-d) = 0$ for $X \leq d$ or $X-d$ for $X > d$

Limit: $Y = \min(X, M) = X$ for $X \leq M$ or M for $X > M$

Useful formula: $E[(X \wedge M)^k] = \int_0^M x^k f_X(x) dx + M^k (1 - F_X(M)) = \sum_{i: x_i \leq M} x_i^k P(X=x_i) + M^k (1 - F_X(M))$

$E[X \wedge M] = \int_0^M (1 - F_X(x)) dx = \sum_{i: x_i \leq M} (1 - F_X(x_i)) (x_{i+1} - x_i)$, $(X-d)_+ = X - (X-d)$

$E[(X-d)_+] = E[X] - E[X \wedge d] = \int_d^\infty (1 - F_X(x)) dx = \sum_{i: x_i > d} (1 - F_X(x_i)) (x_{i+1} - x_i)$

Stop Loss Premium: $P_d = E[(X-d)_+] = P(X>d) e(d)$, $e(d) = E[X-d | X>d]$, $P_d = (E[X-d | X>d])_{pd}$

if claim has both deductible d and max. limit M : $Y = \min(\max(X-d, 0), M)$:

$= \begin{cases} 0, & X \leq d \\ X-d, & d < X < d+M \\ (X-d)_+, & X > d+M \end{cases} = \begin{cases} 0, & X \leq d \\ X-d, & d < X < d+M \\ (X-d)_+, & X > d+M \end{cases}$

Reinsurance:

Proportional: insurer = $Y = \alpha X$, reinsurer = $Z = (1-\alpha)X$, $M_Y = \alpha M_X$, $\sigma_Y^2 = \alpha^2 \sigma_X^2$, $S_Y = S_X$

Non-Proportional: excess of loss: $Y = \min(X, d)$, $Z = \max(X-d, 0)$, reinsurer could apply limit $M \Rightarrow$

limit $M \Rightarrow Y = \min(X, d) + \max(X-M-d, 0)$, $Z = \min(\max(X-d, 0), M)$. stop loss: let $S = \sum_i X_i$

$Y = \min(S, d)$, $Z = \max(S-d, 0)$

M4 - Approximations for Compound Distributions

Panjer Distributions ($a, b, 0$): $P(N=k) = (a + \frac{b}{k}) P(N=k-1)$, or $\frac{P_k}{P_{k-1}} = \frac{a+b}{k}, k=1, 2, \dots$

$\begin{array}{ccccc} a & b & & & P(N=0) \\ \text{Pois}(\lambda) & 0 & \lambda & e^{-\lambda} & \\ \text{Neg Bin}(r, p) & p & (r-1)p & (1-p)^r & \\ \text{Bin}(v, p) & -p/(1-p) & (v+p)/(1-p) & (1-p)^v & \end{array}$

Panjer Distribution ($a, b, 1$): $\frac{P_k}{P_{k-1}} = a + \frac{b}{k}$, $k=2, 3, \dots$, P_0 can be any $p_0 \in [0, 1]$

Zero-truncated dist.: $P_k^T = 0$ for $k=0$ or $\frac{P_k}{P_{k-1}} = \frac{1-p}{1-p_0}$, $k=1, 2, \dots$, P_k from $(a, b, 0)$

Zero-modified dist.: $P_k^M = P_k^0$ for $k=0$ or $\frac{1-P_k}{1-P_0} = P_k$, $k=1, 2, \dots$

Panjer Recursion: $S = \sum_{i=1}^N Y_i$, N is a (a, b) dist., $Y_i \geq 0$ i.i.d. with pmf $g_i = P(Y_i=j)$

$f_S(s) = \frac{1}{1-a g_0} \sum_{j=0}^s (a+b \cdot \frac{j}{s}) g_j f_S(s-j)$, $s=1, 2, \dots$, $f_S(0) = P(N=0)$ or

$M_N(\log(g_0))$ if $g_0 > 0$, e.g. comp. pois. $f_S(s) = \frac{1}{s} \sum_{j=1}^s j g_j f_S(s-j)$, $f_S(0) = e^{\lambda(g_0-1)}$

Normal Approx.: $P(S \leq s) = \Phi\left(\frac{s - E[S]}{\sigma_S}\right)$

$\hookrightarrow S \stackrel{\text{approx}}{\sim} \text{CompPois}(\lambda v, G)$ as $v \rightarrow \infty$, only good for small claims

Translated Gamma/LN Approximation: $X = k + Z$, $Z \sim \text{LN}(\mu, \sigma^2) \sim T(\gamma, c)$

match expected value, variance, and skew.

e.g. fit translated gamma to $\text{CompPois}(\lambda v, G(y)) \stackrel{d}{=} S$, need to solve:

$E[S] = \lambda v E[Y] = k + \gamma/c$, $\text{Var}(S) = \lambda v E[Y^2] = \gamma^2/c^2$, $S = \frac{\lambda v \gamma^2}{(\lambda v)^{1/2} E[Y^{1/2}]} = \frac{\gamma^2}{\sqrt{\lambda v}}$, $\gamma = \left(\frac{2(\lambda v)^{1/2} E[Y^{1/2}]}{E[Y^2]} \right)^{1/2}$, $c = \left(\frac{\gamma}{\lambda v E[Y]} \right)^{1/2}$, $k = \lambda v E[Y] - \frac{\gamma}{c}$ ($\text{also } E[S^2] = \lambda E[Y^2]$)

e.g. translated LN: $E[k + e^{Z \frac{\sigma^2}{2}}] = k + e^{\mu + \frac{\sigma^2}{2}}$, $\text{Var} = (e^{\sigma^2}-1)e^{2\mu+\sigma^2}$, $\text{Skew} = (e^{\sigma^2}+2)\sqrt{e^{\sigma^2}-1}$

M6 - Extreme Value Theory

Measures of Tail Weight: Mean Excess Function: if linear increase \Rightarrow heavy tail. Existence of Moments: Fewer finite moments \Rightarrow heavier tails. Limiting Density Survival Ratio: $\lim_{x \rightarrow \infty} S_A(x) = \lim_{x \rightarrow \infty} f_B(x)$ if ∞ A has heavier tail, if 0 A has lighter tail, if c tails are comparable. Hazard Rate Function: $h(x) = \frac{f(x)}{1-F(x)}$, if increasing in $x \Rightarrow$ light-tailed, if decreasing in $x \Rightarrow$ heavy tailed, if constant \Rightarrow exponential tail.

Block Maxima Notation: $X_{n,n} = \min(X_1, \dots, X_n)$, $X_{1,n} = \max(X_1, \dots, X_n)$, $X_{n,n} \leq X_{n+1,n} \leq \dots \leq X_{\infty,n}$. Gumbel catf

Limiting Argument: (if $X_i \sim \exp(\lambda)$) $P(X_1, \dots, X_n \leq \frac{x+\log(n)}{\lambda}) \approx \Lambda(x) \Rightarrow P(X_{1,n} \leq x) \approx \Lambda(\lambda \cdot \log n)$

GEV: $H_{Y_i; M_i, \sigma}(x) = \begin{cases} e^{-(1+r \cdot \frac{x-M}{\sigma})^{-1/r}}, & r > 0 \\ e^{-e^{-\frac{x-M}{\sigma}}}, & r = 0 \\ e^{-\frac{x-M}{\sigma}}, & r < 0 \end{cases}$ Gumbel (middle size tail) $x \in \mathbb{R}$

e.g. distributions: $r=0$: chi-squared, exponential, gamma, log normal, normal, $r>0$: Burr, F, log-gamma, Pareto, t, beta, uniform, triangular, weibull

Steps: Find a_n and b_n : $\lim_{n \rightarrow \infty} P(\frac{X_{1,n} - b_n}{a_n} \leq x) = \lim_{n \rightarrow \infty} (F(a_n x + b_n))^n$ to match range of x : $x \geq \lim_{n \rightarrow \infty} (-\frac{b_n}{a_n})$, $n(1-F(a_n x + b_n)) = T(x)$, where $T(x) = e^{-T(x)}$ a_n/b_n

For Frechet ($r>0$): limiting distribution of normalised maximum has $r>0$ if: $\bar{F}(x) = 1 - F(x) = x^{-\frac{1}{r}}$, $\lim_{x \rightarrow \infty} L(x) = 1$, $t > 0$ (also $E[X^k] = \infty$ for $k > r$)

For Gumbel ($r=0$): limiting distribution of normalised maximum has $r=0$ if:

$$E[X^k] < \infty, \forall k > 0$$

For Weibull ($r<0$): limiting distribution of normalised maximum has $r<0$ if: it has a finite endpoint ($x_F < \infty$, where $x_F = \sup\{x \in \mathbb{R} : F(x) < 1\}$)

GPD Asymptotic Tail Behaviour: if pricing EoL, retention limit is the t -year quantile: $u_t = F^{-1}(1-\frac{t}{\tau})$. Threshold Exceedance: $F_u(x) = \frac{F(x-u)-F(u)}{1-F(u)} = P(Y-u \leq x | Y > u)$

GPD: $G_{Y,\sigma}(x) = \begin{cases} 1 - (1+r \cdot \frac{x}{\sigma})^{-1/r}, & r < 0 \\ x \in (0, \sigma/|r|) \\ 1 - e^{-\frac{x}{\sigma}}, & r = 0 \\ x \geq 0, & r > 0 \end{cases}$ Pareto (heavy tail) $x \geq 0$

GPD Central Moment (in excess of u): $E[X] = \frac{\sigma}{1-r}, r < 1$, $\text{Var}(X) = \frac{(1-r)^2(1-2r)}{r^2}, r < \frac{1}{2}$, $\hookrightarrow E[X^k] < \infty$ if $r < \frac{1}{k}$, $E[X]$ is the stop loss premium (excess over u) since X is the

Choice of u : We need to estimate tail $\bar{F}(u+x) = \bar{F}(u)\bar{F}_u(x)$, for a fixed large u and $x \geq 0$. $P(\text{exceeding } u) = \hat{F}(u) = \frac{n_u}{n}$, then: $\hat{F}_u(x) = \bar{G}_{Y,\sigma}(u, \hat{\sigma}(u))(x)$. Limitations: Larger u is better approximation, but it reduces amount of data to estimate $\bar{F}(u), r, \sigma$.

Distributions/Formulas

Gamma (α, β): $g(x) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}, E = \frac{\alpha}{\beta}, V = \frac{\alpha}{\beta^2}, S = \sqrt{\frac{2}{\alpha}}$ (pos. skew), $M_x(t) = \left(\frac{\beta}{\beta-t}\right)^\alpha$ ($t < \beta$), $E[X^k] = \frac{\Gamma(\alpha+k)}{\Gamma(\alpha) \beta^k}$

Inverse Gaussian $IG(\alpha, \beta)$: $g(x) = \frac{\alpha^{\alpha/2}}{\sqrt{2\pi\beta^3}} e^{-\frac{(\alpha-\beta x)^2}{2\beta x}}$ ($x > 0$), $E = \frac{\alpha}{\beta}, V = \frac{\alpha}{\beta^2}, S = \sqrt{\frac{3}{\alpha}}$, $M_x(t) = e^{\alpha(1-\sqrt{1-(2t/\beta)})}$ ($t < \beta/2$)

Weibull (γ, c): $g(x) = (c\gamma)^{-1} (cx)^{\gamma-1} e^{-(cx)^\gamma}$ ($x > 0$), $G(x) = 1 - e^{-(cx)^\gamma}$, $E = \frac{\Gamma(1+1/\gamma)}{c}, V = \frac{\Gamma(1+2/\gamma)}{c^2} - \frac{1}{\gamma^2}, S = \left[\frac{\Gamma(1+3/\gamma)}{c^3} - 3M\sigma^2 - M^3\right]/\sigma^3$, $E[X^k] = \frac{\Gamma(1+k/\gamma)}{c^k}$, if $z \sim \exp(1)$, $\Rightarrow z^{1/\gamma} \sim W(\gamma)$

$X \sim LN(\mu, \sigma^2)$: $\log(X) \sim N(\mu, \sigma^2)$, $E = e^{\mu + \frac{\sigma^2}{2}}$, $V = e^{2\mu + \sigma^2} (e^{\sigma^2} - 1)$, $S = (e^{\sigma^2} + 2)(e^{\sigma^2} - 1)$

Log-gamma ($\log X \sim IG(\alpha, \beta)$): $g(x) = \frac{c^\gamma}{\Gamma(\gamma)} (\log(x))^{\gamma-1} x^{-(c+1)}$ ($x > 0$), $E = \left(\frac{c}{c+1}\right)^\gamma$ ($c > 1$), $V = \left(\frac{c}{c+1}\right)^2 - \frac{1}{c+1}$, $S = \left[\left(\frac{c}{c+1}\right)^3 - 3M\sigma^2 - M^3\right]/\sigma^3$, $E[X^k] = \left(\frac{c}{c+1}\right)^k$ ($c > k$)

Pareto (θ, α): $g(x) = \frac{\alpha}{\theta} \left(\frac{x}{\theta}\right)^{-\alpha-1}$ ($x > \theta$), $G(x) = 1 - \left(\frac{\theta}{x}\right)^\alpha$, $E = \theta \cdot \frac{\alpha}{\alpha-1}$ ($\alpha > 1$), $V = \theta^2 \cdot \frac{\alpha}{(\alpha-1)^2(\alpha-2)}$ ($\alpha > 2$), $S = \frac{2(\alpha-1)}{\alpha-3} \left(\frac{\alpha-2}{\alpha}\right)^{1/2}$ ($\alpha > 3$). Let $Z = Y - \theta$: $F_Z(z) = 1 - \left(\frac{\theta}{\theta+z}\right)^\alpha$

Definition of e^α : $\lim_{n \rightarrow \infty} ((1 + \frac{\alpha}{n})^n) = e^\alpha$ or $\lim_{n \rightarrow 0} ((1 - \alpha n)^{\frac{1}{n}}) = e^\alpha$

MGF: $M_X(t) = E[e^{xt}]$ **PGF:** $P_X(t) = E[t^X]$ **CGF:** $K_X(t) = \log M_X(t)$, $E[X] = K'_X(0) = \frac{V_{xx}(0)}{K''_X(0)}$

If $Y = G(X)$: $F_Y(x) = F_X(G^{-1}(y))$ for $G \uparrow$ or $1 - F_X(G^{-1}(y))$ for $G \downarrow$. $f_Y(y) = f_X(G^{-1}(y)) \frac{1}{|G'(G^{-1}(y))|}$

M5 - Copulas

Pearson's Corr. Coef.: $P = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$. Kendall's Tau: $\tau(Z_i, Z_j) = \text{P}((Z_i - Z'_i)(Z_j - Z'_j) > 0) - \text{P}((Z_i - Z'_i)(Z_j - Z'_j) < 0)$. Spearman's Rho: $Z_i \sim F_i, Z_j \sim F_j$, $r(Z_i, Z_j) = P(F_i(Z_i) < F_j(Z_j))$. Sklar's Theorem: $\exists C(x_1, \dots, x_n) = C(F_1(x_1), \dots, F_n(x_n))$

Copula Conditions: non-decreasing, right continuous. ① $\lim_{u_k \rightarrow 0} C(u_1, u_2) = 0, k=1,2$ $\forall u_1, u_2$, ② $\lim_{u_k \rightarrow 1} C(u_1, u_2) = u_1, k=1,2$ ③ $C(v_1, v_2) - C(u_1, v_2) - C(v_1, u_2) + C(u_1, u_2) > 0, \forall u_1, v_1, v_2$

Invariance Property: if \vec{X} has copula C , and T_1, \dots, T_n are strictly increasing $\Rightarrow (T_1(X_1), \dots, T_n(X_n)) \sim \text{copula } C$, so copula holds under log, inflation, etc.

Gaussian Copula: $C(u_1, \dots, u_n) = \Phi_2(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_n))$ or $C(u_1, u_n) = \Phi_p(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_n))$

Fréchet Bounds: all copulas satisfy $L_F(u_1, \dots, u_n) \leq C(u_1, \dots, u_n) \leq U_F(u_1, \dots, u_n)$, where $L_F = \max(0, \sum_{k=1}^n u_k - (n-1))$, $U_F = \min(u_1, \dots, u_n)$, $\lambda_L = \lambda_U = 1$

Comonotonicity Copula: $C(u, v) = \min(u, v)$, where the R.V.'s are perf. pos. dependent

Countermonotonicity Copula: $C(u, v) = \max(u+v-1, 0)$, where R.V.'s are perf. neg. dependent $\lambda_L = \lambda_U = 0$

Survival Copulas: $\bar{F}(x_1, x_2) = \text{P}(X_1 > x_1, X_2 > x_2) = \bar{C}(\bar{F}_1(x_1), \bar{F}_2(x_2))$, where

$\bar{C}(1-u_1, 1-u_2) = 1 - u_1 - v + C(u_1, u_2)$, since $\bar{F}(x_1, x_2) = 1 - F_1(x_1) - F_2(x_2) + F(x_1, x_2)$

Coef. of Lower Tail Dependence: $\lambda_L = \lim_{u \rightarrow 0^+} \text{P}(X_1 \leq F_1^{-1}(u) | X_2 \leq F_2^{-1}(u)) = \lim_{u \rightarrow 0^+} \frac{C(u, u)}{u} \in [0, 1]$

Coef. of Upper Tail Dependence: $\lambda_U = \lim_{u \rightarrow 1^-} \text{P}(X_1 > F_1^{-1}(u) | X_2 > F_2^{-1}(u)) = \lim_{u \rightarrow 1^-} \frac{\bar{C}(1-u, 1-u)}{1-u} = \lim_{u \rightarrow 0^+} \frac{\bar{C}(u, u)}{u} \in [0, 1]$ ($\lambda = 0$ (no dependence), $\lambda = 1$ (full dependence))

Archimedean Copulas: if $C(u_1, \dots, u_n) = \Psi^{-1}(\Psi(u_1) + \dots + \Psi(u_n))$, Ψ = "generator", $\Psi'(1) = 0$, Ψ is strictly decreasing ($\Psi'(u) < 0, \forall u$), Ψ is convex ($\Psi''(u) \geq 0, \forall u$).

Kendall's tau relation: $\tau = 1 + 4 \int_0^1 \frac{\Psi(t)}{\Psi'(t)} dt$

Clayton Copula: $C(u_1, u_2) = (u_1^{-\theta} + u_2^{-\theta} - 1)^{-\frac{1}{\theta}}$, $\theta \in (0, \infty)$, $\Psi(t) = \frac{1}{\theta}(t^{-\theta}-1), \Psi'(s) = (1+\theta s)^{-\frac{1}{\theta}}$, $\tau = \frac{\theta}{2+\theta} \Leftrightarrow \theta = \frac{2\tau}{1-\tau}$, $\lambda_L = \lim_{u \rightarrow 0^+} \frac{(2-u^{-\theta})^{1/\theta}}{u} = \lim_{u \rightarrow 0^+} \frac{(2-u^{\theta})^{-1/\theta} u}{u} = 2^{-\frac{1}{\theta}}$, $\lambda_U = \lim_{u \rightarrow 1^-} \frac{1-2u+(2u^{\theta})^{-1/\theta}}{1-u} = \lim_{u \rightarrow 1^-} \frac{1-2u+2^{-\frac{1}{\theta}}}{1-u} = 2 - 2^{-\frac{1}{\theta}}$

Frank Copula: $C(u_1, u_2) = -\frac{1}{\theta} \log \left(1 + \frac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{e^{-\theta} - 1} \right)$, $\theta \in \mathbb{R} \setminus \{0\}$, $\Psi(t) = -\log(e^{-\theta t} - 1)$, $\Psi'(s) = -\frac{1}{\theta} \log(1 + e^{-\theta(s-1)})$, $\tau = 1 - \frac{4}{\theta} \int_0^{\theta} \int_0^{\theta} \frac{t}{s} e^{-\theta s} ds dt$, $\lambda_U = \lambda_L = 0 \Rightarrow$ Copula is symmetric

Gumbel Copula: $C(u_1, u_2) = e^{-[-(c \log u_1)^\theta + (-\log u_2)^\theta]/\theta}$, $\theta \in [1, \infty)$, $\Psi(t) = (-\log t)^\theta$, $\Psi'(s) = e^{-t^\theta}$, $\tau = \frac{\theta-1}{\theta} \Leftrightarrow \theta = \frac{1}{1-\tau}$, $\lambda_L = 0$, $\lambda_U = 2 - 2^{-\frac{1}{\theta}}$

Poisson (λ): $f_X(x) = \frac{\lambda^x e^{-\lambda}}{x!}, E = V = \lambda, G = \frac{1}{\lambda} \sum_{k=0}^x \lambda^k e^{-\lambda}$, $M_X(t) = e^{\lambda(e^t-1)}$

NegBinom (r, p): $f_X(x) = \binom{x+r-1}{x} (1-p)^x p^r, E = \frac{rp}{1-p}, V = \frac{rp(1-p)}{(1-p)^2}, M_X(t) = \left(\frac{p}{1-(1-p)e^t}\right)^r$

$t < -\log(1-p)$, $S = \frac{2-p}{\sqrt{1-p}e^t}$, # failures k before observing r successes, with prob. p

Binom (n, p): $f_X(x) = \binom{n}{x} p^x (1-p)^{n-x}$, $E = np, V = np(1-p)$, $S = \frac{1-2p}{\sqrt{np(1-p)}}$, $M_X(t) = (1-p+pe^t)^n$

CompPois (E, V, G): $N \sim \text{Pois}(\lambda)$, $M_X(t) = E[e^{xt}] = \mathbb{E}[X^k] = \lambda^k k!$, $V(S) = \lambda m_2$, $S_c = \mathbb{E}\left[\left(\frac{S-m_2}{\sigma}\right)^2\right] = \frac{\lambda m_3}{(\lambda m_2)^2}$

Taylor Series: $e^{-x} = \sum_{n=0}^{\infty} \frac{(-x)^n}{n!} = 1 - x + \dots$, $\log(1-x) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1} x^n}{n} = x - \frac{x^2}{2} + \dots$, $|x| < 1$

Integration by Parts: $\int u dv = uv - \int v du$

12 Appendix B – Detailed Proofs and Derivations

12.1 Proof of Sklar's Theorem (Sketch)

Let (X, Y) have continuous marginals F_X, F_Y . Define $U = F_X(X)$, $V = F_Y(Y)$. Then $U, V \sim \text{Uniform}(0, 1)$. Define copula:

$$C(u, v) = \Pr(U \leq u, V \leq v) = \Pr(F_X(X) \leq u, F_Y(Y) \leq v).$$

Since $\{x : F_X(x) \leq u\} = \{x : x \leq F_X^{-1}(u)\}$,

$$C(u, v) = \Pr(X \leq F_X^{-1}(u), Y \leq F_Y^{-1}(v)) = F_{X,Y}(F_X^{-1}(u), F_Y^{-1}(v)).$$

Conversely, if C is a copula and F_X, F_Y are marginals, define:

$$F_{X,Y}(x, y) = C(F_X(x), F_Y(y)).$$

One verifies that:

$$F_{X,Y}(x, \infty) = C(F_X(x), 1) = F_X(x), \quad F_{X,Y}(\infty, y) = C(1, F_Y(y)) = F_Y(y),$$

using uniform marginals property of C . Hence the joint CDF has marginals F_X, F_Y .

12.2 Derivation of Panjer Recursion (Detailed)

(See the detailed steps in Module 4, but here we verify coefficient manipulation.)

12.3 Derivation of Special Cases of Archimedean Copulas

12.3.1 Clayton Copula

- Direct limit of $C_\theta(u, v)$:

$$C_\theta(u, v) = (u^{-\theta} + v^{-\theta} - 1)^{-1/\theta} = \exp\left[-\frac{1}{\theta} \ln(u^{-\theta} + v^{-\theta} - 1)\right].$$

As $\theta \rightarrow 0$, use $u^{-\theta} = e^{-\theta \ln u} = 1 - \theta \ln u + o(\theta)$ and similarly for v :

$$u^{-\theta} + v^{-\theta} - 1 = 1 - \theta(\ln u + \ln v) + o(\theta), \quad -\frac{1}{\theta} \ln(\dots) \rightarrow \ln u + \ln v,$$

so

$$\lim_{\theta \rightarrow 0} C_\theta(u, v) = \exp(\ln u + \ln v) = uv.$$

As $\theta \rightarrow \infty$, one of $u^{-\theta}, v^{-\theta}$ dominates. If $M = \max(-\ln u, -\ln v)$,

$$u^{-\theta} + v^{-\theta} - 1 \sim e^{\theta M}, \quad -\frac{1}{\theta} \ln(\dots) \rightarrow -M,$$

giving

$$\lim_{\theta \rightarrow \infty} C_\theta(u, v) = e^{-M} = \min(u, v).$$

- Via lower-tail dependence:

$$\lambda_L(\theta) = \lim_{u \rightarrow 0^+} \frac{C_\theta(u, u)}{u} = \lim_{u \rightarrow 0^+} \frac{(2u^{-\theta} - 1)^{-1/\theta}}{u} = 2^{-1/\theta}.$$

Hence $\theta \rightarrow 0^+ \implies \lambda_L \rightarrow 0$ (independence), and $\theta \rightarrow \infty \implies \lambda_L \rightarrow 1$ (comonotonic).

- Via Kendall's τ :

$$\tau(\theta) = \frac{\theta}{\theta + 2}.$$

Thus $\theta \rightarrow 0 \implies \tau \rightarrow 0$ (independence), $\theta \rightarrow \infty \implies \tau \rightarrow 1$ (comonotonic), and extending to $\theta \rightarrow (-1)^+ \implies \tau \rightarrow -1$ (countermonotonic in the bivariate case).

12.3.2 Frank Copula

(i) Independence as $\theta \rightarrow 0$:

$$\begin{aligned} C(u, v) &= -\frac{1}{\theta} \ln \left(1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1} \right), \\ e^{-\theta u} &= 1 - \theta u + o(\theta), \quad e^{-\theta v} = 1 - \theta v + o(\theta), \quad e^{-\theta} = 1 - \theta + o(\theta), \\ \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1} &= \frac{(-\theta u + o(\theta))(-\theta v + o(\theta))}{-\theta + o(\theta)} = -\theta uv + o(\theta), \\ 1 + \frac{\dots}{\dots} &= 1 - \theta uv + o(\theta), \quad \ln(1 - \theta uv + o(\theta)) = -\theta uv + o(\theta), \\ C(u, v) &= -\frac{1}{\theta}(-\theta uv + o(\theta)) = uv + o(1) \longrightarrow uv. \end{aligned}$$

(ii) Comonotonicity as $\theta \rightarrow +\infty$: Write

$$C(u, v) = -\frac{1}{\theta} \ln \left(1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1} \right).$$

Set $A = e^{-\theta u}$, $B = e^{-\theta v}$, $\delta = e^{-\theta}$. Then as $\theta \rightarrow \infty$,

$$A, B, \delta \rightarrow 0, \quad \frac{(A-1)(B-1)}{\delta-1} = \frac{1+o(1)}{-1+o(1)} = -1+o(1),$$

so the argument of \ln is $1 + (-1 + o(1)) = o(1)$. More precisely one shows

$$C(u, v) = u - \frac{1}{\theta} \ln(1 + e^{-\theta(v-u)}).$$

If $u \leq v$ then $e^{-\theta(v-u)} \rightarrow 0$, hence $C(u, v) \rightarrow u$; similarly if $v \leq u$, $C(u, v) \rightarrow v$. Thus

$$\lim_{\theta \rightarrow +\infty} C(u, v) = \min(u, v).$$

(iii) Countermonotonicity as $\theta \rightarrow -\infty$: Let $\phi = -\theta \rightarrow +\infty$. Then

$$C(u, v) = \frac{1}{\phi} \ln \left(1 + \frac{(e^{\phi u} - 1)(e^{\phi v} - 1)}{e^\phi - 1} \right).$$

An analogous argument shows that for large ϕ ,

$$C(u, v) \longrightarrow u + v - 1,$$

and hence (restricted to $[0, 1]$) the Fréchet lower bound $\max(u + v - 1, 0)$.

12.3.3 Gumbel Copula

Recall

$$C_\theta(u, v) = \exp \left(-[(-\ln u)^\theta + (-\ln v)^\theta]^{1/\theta} \right), \quad \theta \in [1, \infty).$$

(i) Independence copula as $\theta \rightarrow 1$:

$$\lim_{\theta \rightarrow 1} C_\theta(u, v) = \exp \left(-[(-\ln u)^1 + (-\ln v)^1]^{1/1} \right) = \exp(-(-\ln u - \ln v)) = \exp(\ln u + \ln v) = uv.$$

(ii) Comonotonicity copula as $\theta \rightarrow \infty$: Set $A = -\ln u$ and $B = -\ln v$, both nonnegative.

Then

$$(A^\theta + B^\theta)^{1/\theta} = \max(A, B) [1 + (A/B)^\theta]^{1/\theta} \longrightarrow \max(A, B) \quad (\theta \rightarrow \infty).$$

Hence

$$\lim_{\theta \rightarrow \infty} C_\theta(u, v) = \exp(-\max(A, B)) = \exp(-\max(-\ln u, -\ln v)) = \min(u, v).$$

12.4 Derivation of Tail Coefficients of Archimedean Copulas

12.4.1 Clayton Copula

$$\begin{aligned} \lambda_L &= \lim_{u \rightarrow 0^+} \frac{C(u, u)}{u} = \lim_{u \rightarrow 0^+} \frac{(2u^{-\theta} - 1)^{-1/\theta}}{u} = \lim_{u \rightarrow 0^+} \frac{(u^{-\theta}(2 - u^\theta))^{-1/\theta}}{u} \\ &= \lim_{u \rightarrow 0^+} \frac{u(2 - u^\theta)^{-1/\theta}}{u} = 2^{-1/\theta}, \\ \lambda_U &= \lim_{u \rightarrow 1^-} \frac{\bar{C}(1-u, 1-u)}{1-u} = \lim_{u \rightarrow 1^-} \frac{1-2u+C(u,u)}{1-u} \quad (\text{both top and bottom } \rightarrow 0) \\ &\stackrel{\text{l'Hôpital}}{=} \lim_{u \rightarrow 1^-} \frac{-2 + \frac{d}{du}(2u^{-\theta} - 1)^{-1/\theta}}{-1} = \lim_{u \rightarrow 1^-} \left(2 - \frac{d}{du}(2u^{-\theta} - 1)^{-1/\theta} \right). \end{aligned}$$

Now compute the derivative:

$$\frac{d}{du}(2u^{-\theta} - 1)^{-1/\theta} = -\frac{1}{\theta}(2u^{-\theta} - 1)^{-1/\theta-1} \cdot 2(-\theta)u^{-\theta-1} = 2(2u^{-\theta} - 1)^{-1/\theta-1}u^{-\theta-1}.$$

At $u = 1$, $2u^{-\theta} - 1 = 1$, so the derivative equals 2. Hence

$$\lambda_U = \lim_{u \rightarrow 1^-} (2 - 2) = 0.$$

12.4.2 Frank Copula

$$\begin{aligned} \lambda_L &= \lim_{u \rightarrow 0^+} \frac{C(u, u)}{u} = \lim_{u \rightarrow 0^+} \frac{-\frac{1}{\theta} \ln\left(1 + \frac{(e^{-\theta u} - 1)^2}{e^{-\theta} - 1}\right)}{u} \\ &= \lim_{u \rightarrow 0^+} \frac{-\frac{1}{\theta} \ln\left(1 + \frac{\theta^2 u^2 + o(u^2)}{e^{-\theta} - 1}\right)}{u} = \lim_{u \rightarrow 0^+} \frac{-\frac{1}{\theta} \left(\frac{\theta^2 u^2}{e^{-\theta} - 1} + o(u^2)\right)}{u} = 0, \\ \lambda_U &= \lim_{u \rightarrow 1^-} \frac{\bar{C}(1-u, 1-u)}{1-u} = \lim_{u \rightarrow 1^-} \frac{1-2u+C(u,u)}{1-u} \quad (\text{apply l'Hôpital}) \\ &= \lim_{u \rightarrow 1^-} \frac{-2 + \frac{d}{du}C(u,u)}{-1} = \lim_{u \rightarrow 1^-} (2 - \frac{d}{du}C(u,u)). \end{aligned}$$

Since

$$\frac{d}{du}C(u, u) = 2 \frac{\partial C}{\partial u}(u, u),$$

and for the Frank copula

$$\frac{\partial C}{\partial u}(u, v) = \frac{e^{-\theta u}(e^{-\theta v} - 1)}{(e^{-\theta} - 1) \left(1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1}\right)},$$

we evaluate at $(u, v) = (1, 1)$:

$$e^{-\theta u} = e^{-\theta}, \quad e^{-\theta v} - 1 = e^{-\theta} - 1, \quad 1 + \frac{(e^{-\theta} - 1)^2}{e^{-\theta} - 1} = e^{-\theta},$$

so

$$\frac{\partial C}{\partial u}(1, 1) = \frac{e^{-\theta}(e^{-\theta} - 1)}{(e^{-\theta} - 1)e^{-\theta}} = 1,$$

hence $\frac{d}{du}C(u, u)|_{u=1} = 2$ and therefore

$$\lambda_U = 2 - 2 = 0.$$

12.4.3 Gumbel Copula

Recall

$$C_\theta(u, v) = \exp\left(-[(-\ln u)^\theta + (-\ln v)^\theta]^{1/\theta}\right).$$

Lower tail dependence λ_L :

$$\begin{aligned} \lambda_L &= \lim_{u \rightarrow 0^+} \frac{C_\theta(u, u)}{u} = \lim_{u \rightarrow 0^+} \frac{\exp(-[2(-\ln u)^\theta]^{1/\theta})}{u} = \lim_{u \rightarrow 0^+} \frac{\exp(-2^{1/\theta}(-\ln u))}{u} \\ &= \lim_{u \rightarrow 0^+} \frac{u^{2^{1/\theta}}}{u} = \lim_{u \rightarrow 0^+} u^{2^{1/\theta}-1} = 0, \quad \text{since } 2^{1/\theta} > 1 \ (\theta \geq 1). \end{aligned}$$

Upper tail dependence λ_U : Use the Taylor expansion of $-\ln u$ near $u = 1$:

$$-\ln u = (1 - u) + o(1 - u).$$

Then

$$C_\theta(u, u) = \exp(-2^{1/\theta}(1 - u + o(1 - u))) = 1 - 2^{1/\theta}(1 - u) + o(1 - u).$$

Hence the survival-copula numerator

$$\bar{C}(1 - u, 1 - u) = 1 - 2u + C_\theta(u, u) = 2(1 - u) - 2^{1/\theta}(1 - u) + o(1 - u) = (2 - 2^{1/\theta})(1 - u) + o(1 - u),$$

and therefore

$$\lambda_U = \lim_{u \rightarrow 1^-} \frac{\bar{C}(1 - u, 1 - u)}{1 - u} = 2 - 2^{1/\theta}.$$

12.5 Derivation of Kendall's Tau of Archimedean Copulas

12.5.1 Clayton Copula

We start from the Archimedean formula

$$\tau = 1 + 4 \int_0^1 \frac{\phi(t)}{\phi'(t)} dt,$$

and use the Clayton generator

$$\phi(t) = \frac{t^{-\theta} - 1}{\theta}, \quad \phi^{-1}(s) = (1 + \theta s)^{-1/\theta},$$

so that

$$C(u, v) = \phi^{-1}(\phi(u) + \phi(v)) = (u^{-\theta} + v^{-\theta} - 1)^{-1/\theta}.$$

Step 1. Compute $\phi'(t)$:

$$\phi'(t) = \frac{d}{dt} \left(\frac{t^{-\theta} - 1}{\theta} \right) = \frac{-\theta t^{-\theta-1}}{\theta} = -t^{-\theta-1}.$$

Step 2. Form the integrand:

$$\frac{\phi(t)}{\phi'(t)} = \frac{\frac{t^{-\theta}-1}{\theta}}{-t^{-\theta-1}} = \frac{t^{-\theta}-1}{\theta} t^{\theta+1} = \frac{t-t^{\theta+1}}{\theta}.$$

Step 3. Plug into the formula for τ :

$$\tau = 1 + 4 \int_0^1 \frac{\phi(t)}{\phi'(t)} dt = 1 + \frac{4}{\theta} \int_0^1 (t - t^{\theta+1}) dt.$$

Split the integral:

$$\int_0^1 t dt = \frac{1}{2}, \quad \int_0^1 t^{\theta+1} dt = \frac{1}{\theta+2}.$$

Hence

$$\int_0^1 (t - t^{\theta+1}) dt = \frac{1}{2} - \frac{1}{\theta+2} = \frac{\theta}{2(\theta+2)}.$$

Thus

$$\tau = 1 + \frac{4}{\theta} \frac{\theta}{2(\theta+2)} = 1 + \frac{2}{\theta+2} = \frac{\theta+2+2}{\theta+2} = \frac{\theta}{\theta+2}.$$

Conclusion. For the Clayton copula with parameter θ ,

$$\tau_{\text{Clayton}}(\theta) = \frac{\theta}{\theta+2}.$$

12.5.2 Gumbel Copula

We start from the Archimedean formula

$$\tau = 1 + 4 \int_0^1 \frac{\phi(t)}{\phi'(t)} dt.$$

For the Gumbel copula with parameter $\theta \geq 1$, the generator is

$$\phi(t) = (-\ln t)^\theta, \quad \phi'(t) = -\frac{\theta(-\ln t)^{\theta-1}}{t}.$$

Hence

$$\frac{\phi(t)}{\phi'(t)} = \frac{(-\ln t)^\theta}{-\frac{\theta(-\ln t)^{\theta-1}}{t}} = \frac{t \ln t}{\theta},$$

and so

$$\tau = 1 + 4 \int_0^1 \frac{\phi(t)}{\phi'(t)} dt = 1 + \frac{4}{\theta} \int_0^1 t \ln t dt.$$

Integration by parts for $\int_0^1 t \ln t dt$:

Let

$$u = \ln t, \quad dv = t dt, \quad du = \frac{1}{t} dt, \quad v = \frac{t^2}{2}.$$

Then

$$\int_0^1 t \ln t dt = \left[\frac{t^2}{2} \ln t \right]_0^1 - \int_0^1 \frac{t^2}{2} \cdot \frac{1}{t} dt = \left[\frac{t^2}{2} \ln t \right]_0^1 - \frac{1}{2} \int_0^1 t dt.$$

Evaluate the boundary term and the remaining integral:

$$\left[\frac{t^2}{2} \ln t \right]_0^1 = \frac{1}{2} \ln 1 - \lim_{t \rightarrow 0^+} \frac{1}{2} \ln t = 0 - 0 = 0, \quad \int_0^1 t dt = \frac{1}{2}.$$

Hence

$$\int_0^1 t \ln t dt = 0 - \frac{1}{2} \cdot \frac{1}{2} = -\frac{1}{4}.$$

Substituting back into our formula for τ :

$$\tau = 1 + \frac{4}{\theta} \left(-\frac{1}{4} \right) = 1 - \frac{1}{\theta},$$

and therefore

$$\boxed{\tau_{\text{Gumbel}}(\theta) = 1 - \frac{1}{\theta}, \quad \theta \geq 1.}$$

12.6 Frank Copula

We start from the Archimedean-copula formula

$$\tau = 1 + 4 \int_0^1 \frac{\varphi(t)}{\varphi'(t)} dt,$$

where for the Frank copula with parameter $\theta \neq 0$,

$$\varphi(t) = \ln(1 - e^{-\theta}) - \ln(1 - e^{-\theta t}), \quad \varphi'(t) = \frac{\theta e^{-\theta t}}{1 - e^{-\theta t}}.$$

1. Define the auxiliary function $H(\theta)$

Let

$$H(\theta) = \int_0^1 \ln(1 - e^{-\theta t}) dt.$$

2. Express the core integral in terms of H

Observe that

$$\varphi(t) = \ln(1 - e^{-\theta}) - \ln(1 - e^{-\theta t}),$$

so

$$\int_0^1 \frac{\varphi(t)}{\varphi'(t)} dt = \int_0^1 \frac{\ln(1 - e^{-\theta}) - \ln(1 - e^{-\theta t})}{\frac{\theta e^{-\theta t}}{1 - e^{-\theta t}}} dt = \frac{\ln(1 - e^{-\theta})}{\theta} - \frac{1}{\theta} H(\theta) - \frac{1}{\theta}.$$

Hence

$$\tau(\theta) = 1 + 4 \left[\frac{\ln(1 - e^{-\theta}) - H(\theta)}{\theta} - \frac{1}{\theta} \right].$$

3. Differentiate $H(\theta)$ under the integral sign

By Leibniz's rule,

$$H'(\theta) = \frac{d}{d\theta} \int_0^1 \ln(1 - e^{-\theta t}) dt = \int_0^1 \frac{\partial}{\partial \theta} \ln(1 - e^{-\theta t}) dt = \int_0^1 \frac{-t e^{-\theta t}}{1 - e^{-\theta t}} dt.$$

Set $s = \theta t$, so $dt = ds/\theta$ and $s \in [0, \theta]$. Then

$$H'(\theta) = -\frac{1}{\theta} \int_0^\theta \frac{s}{e^s - 1} ds,$$

which gives

$$\int_0^\theta \frac{s}{e^s - 1} ds = -\theta H'(\theta).$$

4. Final assembly

Substitute back into the expression for $\tau(\theta)$:

$$\tau(\theta) = 1 + 4 \left[\frac{\ln(1 - e^{-\theta}) - H(\theta)}{\theta} - \frac{1}{\theta} \right] = 1 - \frac{4}{\theta} + \frac{4}{\theta^2} \int_0^\theta \frac{s}{e^s - 1} ds.$$

This completes the proof of the integral form

$$\boxed{\tau(\theta) = 1 - \frac{4}{\theta} + \frac{4}{\theta^2} \int_0^\theta \frac{s}{e^s - 1} ds}.$$

12.7 Derivation of GEV as Limit of Maxima

Outline:

1. Let F_X be distribution of i.i.d. X_i . Suppose that for some sequences $a_n > 0$, b_n , $\Pr((M_n - b_n)/a_n \leq z) = F_X^n(a_n z + b_n) \rightarrow G(z)$.
2. Show G must be of one of three extreme value types: Gumbel, Fréchet, or Weibull. This involves solving functional equation for possible limit distributions.
3. Parameterize G in unified form:

$$G(z) = \exp\left(-\left[1 + \xi \frac{z - \mu}{\sigma}\right]_+^{-1/\xi}\right).$$

12.8 Derivation of ACF for AR(1) and MA(1)

12.8.1 AR(1)

$$Y_t = \phi Y_{t-1} + Z_t, \quad |\phi| < 1.$$

Stationary solution: $Y_t = \sum_{j=0}^{\infty} \phi^j Z_{t-j}$. Then

$$\begin{aligned} \gamma(h) &= \text{Cov}\left(\sum_{j=0}^{\infty} \phi^j Z_{t-j}, \sum_{k=0}^{\infty} \phi^k Z_{t+h-k}\right) \\ &= \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} \phi^j \phi^k \text{Cov}(Z_{t-j}, Z_{t+h-k}). \end{aligned}$$

Because $\text{Cov}(Z_{t-j}, Z_{t+h-k}) = 0$ unless $t - j = t + h - k$, i.e. $k = j + h$.

$$= \sum_{j=0}^{\infty} \phi^j \phi^{j+h} \sigma^2 = \phi^h \sigma^2 \sum_{j=0}^{\infty} \phi^{2j} = \phi^h \frac{\sigma^2}{1 - \phi^2}.$$

In particular, $\gamma(0) = \sigma^2/(1 - \phi^2)$. Hence

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \phi^h, \quad h = 0, 1, 2, \dots$$

12.8.2 MA(1)

$$Y_t = Z_t + \theta Z_{t-1}.$$

Compute $\gamma(0)$:

$$\gamma(0) = \text{Var}(Z_t + \theta Z_{t-1}) = \sigma^2(1 + \theta^2).$$

For $h = 1$:

$$\gamma(1) = \text{Cov}(Z_t + \theta Z_{t-1}, Z_{t-1} + \theta Z_{t-2}) = \sigma^2\theta.$$

For $h > 1$, $\gamma(h) = 0$. Hence

$$\rho(1) = \frac{\theta}{1 + \theta^2}, \quad \rho(h) = 0 \text{ for } h > 1.$$

12.9 Predictions and Graphs of log data SARIMA

Example with jj data:

```
fit.gg <- sarima(log(jj), 1, 1, 0, 1, 1, 0, 4) # also sMA(1) instead of sAR(1) is good
jj.pr <- predict(fit.jj$fit, n.ahead = 40, ci = 0.95)
jj.pred <- ts(c(jj, exp(jj.pr$pred)), start = start(jj), frequency = frequency(jj))

# Plot
ts.plot(jj.pred, ylim = c(0, 60))
U = exp(jj.pr$pred + jj.pr$se)
L = exp(jj.pr$pred - jj.pr$se)
xx = c(time(U), rev(time(U)))
yy = c(L, rev(U))
polygon(xx, yy, border = 8, col = gray(0.6, alpha = 0.2))
lines(exp(jj.pr$pred), col = "red", lwd = 1, type = "p")
```

12.10 True and False Statements for Time Series

1. The theoretical autocorrelation function (ACF) of a pure MA(q) process cuts off after lag q .
 - **True:** By definition, for an MA(q) the autocovariances $\gamma(h) = 0$ for all $h > q$, so the ACF is zero beyond lag q .
2. The theoretical partial autocorrelation function (PACF) of a pure AR(p) process cuts off after lag p .
 - **True:** In an AR(p), the PACF $\alpha(h) = 0$ for all $h > p$ in the population, since lags beyond p carry no additional direct information.
3. A stationary ARMA(p, q) process is causal if its AR characteristic polynomial's roots lie outside the unit circle.
 - **True:** Causality requires that x_t can be written as $x_t = \sum_{j \geq 0} \psi_j w_{t-j}$ with $\sum |\psi_j| < \infty$, which holds when the AR roots are all $|z| > 1$.

4. A stationary ARMA(p, q) process is invertible if its MA characteristic polynomial's roots lie outside the unit circle.
 - **True:** Invertibility means w_t can be expressed as an MA(∞) in x_t , requiring MA-side roots $|z| > 1$.
5. White noise has zero autocorrelation at all nonzero lags.
 - **True:** By definition $\gamma(h) = 0$ for $h \neq 0$ for a white-noise sequence.
6. The sample ACF at lag h is an unbiased estimator of the true autocorrelation $\rho(h)$.
 - **False:** The sample ACF is biased, especially for large h and small samples, because it uses the same mean estimate in all lags.
7. The Yule–Walker equations always yield unbiased estimates of AR(p) coefficients when solved on the sample ACF.
 - **False:** Estimates obtained by plugging the sample ACF into Yule–Walker are biased in finite samples.
8. For an AR(p) with characteristic polynomial $\phi(z)$, stationarity is equivalent to $\phi(z) \neq 0$ for all $|z| \leq 1$.
 - **False:** The correct condition is $\phi(z) \neq 0$ for all $|z| \leq 1$ *including* the unit circle; roots must lie strictly *outside* $|z| = 1$.
9. Differencing a non-stationary time series always produces a stationary series.
 - **False:** Over-differencing can induce MA structure and may still be non-stationary if an underlying trend remains.
10. The Box–Jenkins modelling cycle is: (i) identification, (ii) estimation, (iii) diagnostic checking, (iv) forecasting.
 - **True:** These are the four canonical steps in ARIMA model building.
11. The MA(∞) coefficients $\{\psi_j\}$ of an ARMA(p, q) are obtained by long-division of $\theta(z)$ by $\phi(z)$.
 - **True:** One writes $\theta(z)/\phi(z) = \sum_{j \geq 0} \psi_j z^j$ and equates coefficients.
12. The theoretical ACF of a stationary AR(1) with $|\phi| < 1$ decays hyperbolically.
 - **False:** It decays *geometrically*: $\rho(h) = \phi^h$.
13. In a stationary AR(2) with complex-conjugate roots outside the unit circle, the ACF decays as a damped sine wave.
 - **True:** Complex roots imply oscillatory behavior with exponential damping.
14. For an ARMA(p, q), the PACF eventually decays exponentially rather than cutting off.
 - **True:** Unless $q = 0$ (pure AR), the PACF has an infinite-lag tail that decays.
15. The innovation variance σ^2 can be obtained by solving the first Yule–Walker equation for $\gamma(0)$.
 - **True:** The equation $\gamma(0) = \sum_{i=1}^p \phi_i \gamma(i) + \sigma^2$ gives $\sigma^2 = \gamma(0) - \sum \phi_i \gamma(i)$.
16. The sample PACF at lag k is exactly zero whenever $k > p$ for an AR(p), in any finite sample.
 - **False:** Only the *theoretical* PACF is zero beyond p ; the sample PACF is only approximately zero due to sampling variability.
17. The variance of the sample ACF estimator at lag h is approximately $1/n$ for large n .
 - **True:** Under white-noise assumptions, $[\hat{\rho}(h)] \approx 1/n$ for fixed h .
18. The Yule–Walker equations yield unbiased estimates of AR(p) coefficients in finite samples.
 - **False:** In finite samples the sample autocovariances introduce bias, so Yule–Walker estimates are biased, especially for small n .
19. The Yule–Walker estimator for AR coefficients is **consistent** as $n \rightarrow \infty$.
 - **True:** Under stationarity and ergodicity, sample autocovariances converge to their population values, so Yule–Walker estimates converge to the true coefficients.
20. The Yule–Walker equations can be used directly to estimate MA(q) coefficients.
 - **False:** Yule–Walker applies only to pure AR(p) models; MA or ARMA require other methods.

21. For a Gaussian AR(p) process, the Yule–Walker estimator coincides with the maximum-likelihood estimator.
- **True:** Under Gaussianity, the likelihood equations reduce to the Yule–Walker equations, so both estimators agree.
22. The Yule–Walker solution exists and is unique if the true autocovariance matrix is positive-definite.
- **True:** Positive-definiteness guarantees the Toeplitz matrix is invertible, yielding a unique solution.
23. The formula $\hat{\phi}_1 = \hat{\gamma}(1)/\hat{\gamma}(0)$ for AR(1) is an unbiased estimator of ϕ_1 .
- **False:** While $[\hat{\gamma}(1)] = (n - 1)\gamma(1)/n$, the ratio $\hat{\gamma}(1)/\hat{\gamma}(0)$ remains biased in finite samples.
24. Yule–Walker estimation requires centering the data by subtracting the sample mean.
- **True:** The equations assume zero-mean data, so you must remove any nonzero mean beforehand.
25. Overfitting the AR order p in Yule–Walker leads to inflated estimation variance but still solves the system.
- **True:** A larger p increases parameter uncertainty (variance), though the linear system remains solvable if the matrix is invertible.
26. In an ARCH(1) model, the unconditional variance (x_t) exists only if $\alpha_1 < 1$.
- **True:** Stationarity of the second moment requires $\alpha_1 < 1$, giving

$$(x_t) = \frac{\alpha_0}{1 - \alpha_1} < \infty.$$

27. In an ARCH(1) process, the returns x_t are uncorrelated, but the squared returns x_t^2 exhibit autocorrelation.
- **True:** ARCH generates “volatility clustering,” so $(x_t, x_{t-k}) = 0$ but $(x_t^2, x_{t-k}^2) \neq 0$ for some $k > 0$.
28. A VAR(p) model is stationary if all eigenvalues of its companion matrix lie strictly within the unit circle.
- **True:** Equivalently, the roots of $\det(I - \Phi_1 z - \cdots - \Phi_p z^p) = 0$ must satisfy $|z| > 1$, which is the same as companion-matrix eigenvalues having modulus < 1 .
29. A stable VAR(p) admits a VMA(∞) representation in which each component is a linear combination of current and past shocks.
- **True:** Under stationarity $(I - \sum_{i=1}^p \Phi_i z^i)^{-1}$ can be expanded as $\sum_{j \geq 0} \Psi_j z^j$.
30. Each equation of a VAR(p) system can be estimated consistently by OLS on that equation alone.
- **True:** Since all right-hand-side regressors are predetermined, OLS is consistent even if the error-covariance matrix is nondiagonal.
31. The theoretical ACF of a pure MA(q) process decays exponentially toward zero as lag $h \rightarrow \infty$.
- **False:** An MA(q) has autocorrelations $\rho(h) = 0$ for all $h > q$, so its ACF cuts off after lag q rather than decays exponentially.
32. Pure seasonal ARMA($P, Q)_s$ models only have nonzero autocorrelations at integer multiples of the seasonal period s .
- **True:** By construction, the autocorrelation $\rho(h) = 0$ unless h is a multiple of s , since only seasonal lags enter the model.
33. Seasonal differencing $(1 - B^s)$ always yields a stationary series, regardless of any nonseasonal trend.
- **False:** Seasonal differencing removes only the seasonal unit root; if a nonseasonal trend remains, further nonseasonal differencing $(1 - B)$ may still be needed.
34. In an ARIMA(p, d, q), the d -th difference $\nabla^d x_t$ is guaranteed to be stationary.

- **True:** By definition, $\nabla^d x_t$ follows an ARMA(p, q) model, which is stationary when its AR roots lie outside the unit circle.
35. The Yule–Walker equations can be used to estimate the parameters of an AR(p) model but not an MA(q).
- **True:** Yule–Walker relates autocovariances to AR coefficients; there is no analogous closed-form system for pure MA models.
36. A pure seasonal MA(1)₁₂ model has a nonzero ACF only at lags $h = 0$ and $h = \pm 12$.
- **True:** The autocovariance is nonzero only at lag 12 (and hence at -12), so the ACF is nonzero only at those lags.
37. An AR(1) model with $\phi = 1$ (a random walk) is both stationary and causal.
- **False:** $\phi = 1$ places a unit root on the unit circle, violating stationarity and preventing a decaying impulse-response, so it is nonstationary and noncausal.
38. The sample ACF alone can always distinguish between pure AR and ARMA models.
- **False:** ARMA models can exhibit slowly decaying ACFs similar to AR; one usually needs the PACF or other diagnostics to differentiate them.
39. In an ARCH(1) model, the conditional variance $(x_t | \mathcal{F}_{t-1})$ depends on $(x_{t-1} - \mu)^2$.
- **True:** By definition, $(x_t | \mathcal{F}_{t-1}) = \alpha_0 + \alpha_1(x_{t-1} - \mu)^2$, so it directly depends on the previous squared deviation.
40. The ARIMA(0,1,1) (IMA(1,1)) model is mathematically equivalent to simple exponential smoothing.
- **True:** The IMA(1,1) model yields forecasts that follow an exponentially weighted moving average (EWMA) with smoothing parameter equal to the MA coefficient.
41. Multiplicative SARIMA models combine seasonal and nonseasonal AR and MA polynomials multiplicatively.
- **True:** A SARIMA(p, d, q) \times (P, D, Q) _{s} model is written as $\Phi_P(B^s) \phi(B) \nabla_s^D \nabla^d x_t = \Theta_Q(B^s) \theta(B) w_t$.
42. In a pure seasonal ARMA(P, Q) _{s} model, the theoretical ACF is nonzero only at lags that are multiples of s , up to $\max(P, Q) \times s$.
- **True:** Seasonal lags beyond $\max(P, Q) s$ have zero autocorrelation.
43. Parameter redundancy in ARMA models occurs if the AR and MA polynomials share a common factor.
- **True:** Multiplying both sides of an ARMA by a factor $\eta(B)$ gives the same process with extra, redundant parameters.
44. The Box–Pierce (Portmanteau) test for VAR residuals is asymptotically chi-square under the null of no residual autocorrelation.
- **True:** Under white-noise residuals, the Portmanteau statistic follows χ^2 with appropriate degrees of freedom.
45. Over-differencing a time series can introduce artificial moving-average structure and may still leave nonstationarity.
- **True:** Applying more differences than necessary can create an MA component and need not guarantee stationarity.
46. The partial autocorrelation at lag h of a differenced series $\nabla^d x_t$ can help identify the AR order of the original ARIMA model.
- **True:** After differencing d times, the PACF of the resulting stationary series cuts off at the AR order p .
47. The best linear predictor (BLP) of x_{n+m} is obtained by solving $E[(x_{n+m} - \hat{x}_n^{n+m})x_k] = 0$ for $k = 0, 1, \dots, n$.
- **True:** These “prediction equations” guarantee the BLP minimizes mean square error.
48. For a causal, invertible ARMA(p, q), the infinite-past forecast $\tilde{x}_{n+m} = E[x_{n+m} | x_n, x_{n-1}, \dots]$ is easier to compute than the finite-sample forecast \hat{x}_n^{n+m} .
- **True:** Conditioning on the infinite past lets us use the MA() or AR() expansion

directly without solving an $n \times n$ system.

49. The Durbin–Levinson algorithm can be used to compute the one-step-ahead forecast coefficients $\{\phi_{n,k}\}$ without ever inverting the full autocovariance matrix.
 - **True:** It recursively builds the $\phi_{n,k}$ and prediction error variances P_n^{n+1} .
50. The one-step-ahead mean-square prediction error is $P_n^{n+1} = \gamma(0) - \gamma'_n \Gamma_n^{-1} \gamma_n$.
 - **True:** This follows directly from the Yule–Walker–type “prediction equations.”
51. In practice, truncating the infinite sums in the ARMA forecast formula (i.e. using only x_1, \dots, x_n) introduces no approximation error for AR(p) models when forecasting any horizon m .
 - **True:** For pure AR(p), the infinite-past and finite-sample predictors coincide once $n > p$, so no truncation is needed.
52. The m -step-ahead forecast error variance for a causal ARMA satisfies

$$P_n^{n+m} = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2,$$

and therefore converges to $\gamma(0)$ exponentially fast as $m \rightarrow \infty$.

- **True:** The ψ_j decay geometrically for a stable ARMA, so $P_n^{n+m} \rightarrow \gamma(0)$ quickly.
53. The infinite-past forecast \tilde{x}_{n+m} and the finite-sample forecast \hat{x}_n^{n+m} are identical for all m when the model is invertible.
 - **False:** They only coincide asymptotically or under extra conditions; in general \hat{x}_n^{n+m} solves a different (finite) system.
 54. The ARIMA(0,1,1) model yields exactly the same point forecasts as simple exponential smoothing with smoothing parameter equal to the MA coefficient.
 - **True:** The IMA(1,1) formulation is algebraically equivalent to an EWMA forecast.
 55. Long-range forecasts from a stable ARMA process always revert exponentially fast to the series mean.
 - **True:** As $m \rightarrow \infty$, $\tilde{x}_{n+m} \rightarrow \mu_x$ at the same geometric rate as the MA() coefficients decay.

13 Appendix C - General R

13.1 R Piping

```
# Pipe operator: pass result from left to right
data %>% function1() %>% function2()

# Select columns
data %>% select(column1, column2, ...)

# Filter rows based on a condition
data %>% filter(condition)

# Create or modify columns
data %>% mutate(new_column = expression)

# Sort rows
data %>% arrange(column)
data %>% arrange(desc(column))

# Group and summarise
data %>%
  group_by(group_column) %>%
  summarise(
    summary1 = summary_function1(),
    summary2 = summary_function2()
  )

# Create block maxima (e.g., max per year)
data %>%
  group_by(block_column) %>%
  summarise(max_value = max(target_column, na.rm = TRUE))
```

13.2 Using polyroot

Using `polyroot` to find roots of polynomials in R (good for finding roots of characteristic polynomials for AR and MA).

```
# Suppose AR(2) with phi1 = 0.5, phi2 = -0.3
phi <- c(0.5, -0.3)

# Build coefficient vector for 1 - phi1*z - phi2*z^2:
coefs <- c(1, -phi) # == c(1, -0.5, 0.3)

roots <- polyroot(coefs)
print(roots)
```

13.3 Installing tinytex

```
install.packages("tinytex")
tinytex::install_tinytex()
```

14 Appendix D - Practice Final Exam Worked Solutions

refer to next page

ACTL30007 AM3 : ACTL90020 GIM

Practice Final Exam

Question 1

The attached dataset `practice_q1.xlsx` contains claims amounts in a travel insurance personal items cover:

- The column `claim` corresponds to the actual amount paid;
- The column `limit` corresponds to the limit applicable to that claim;
- The column `deductible` corresponds to the deductible applicable to that claim.

Note that if `claim=limit` then you can assume that the claim has been censored at that level, that is, the claim has reached its applicable limit.

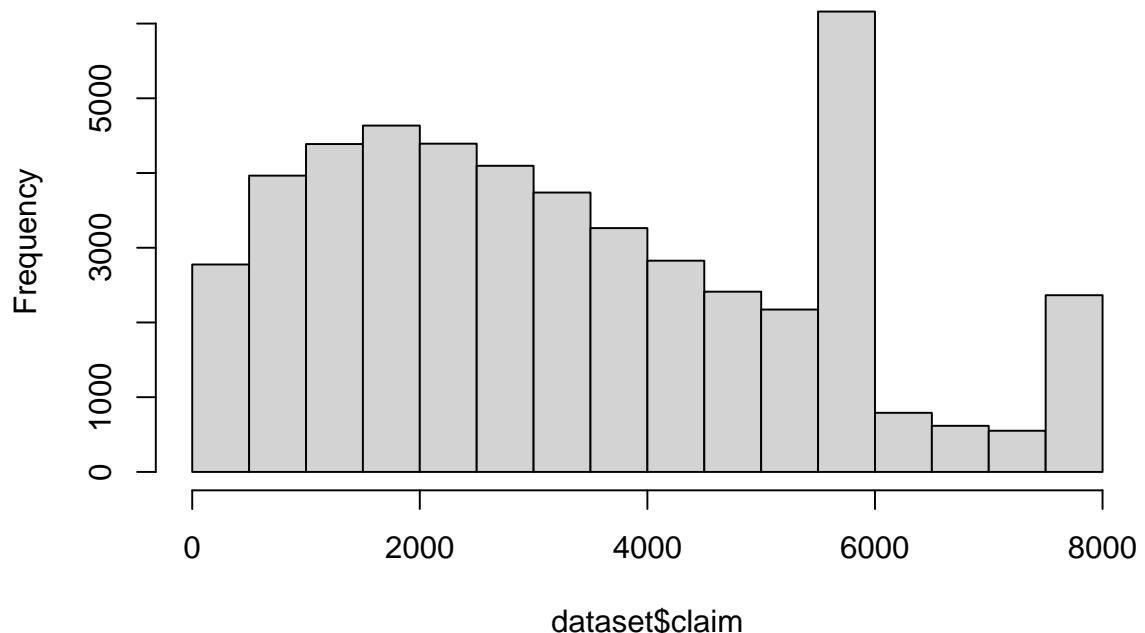
Perform the following tasks:

- (a) Provide two histograms: one for the claims, and one for the raw (before the deductible is applied) damage costs, which could be censored. Explain which of their features are due to the existence of deductibles and limits. (2 marks)

The histograms are as follows:

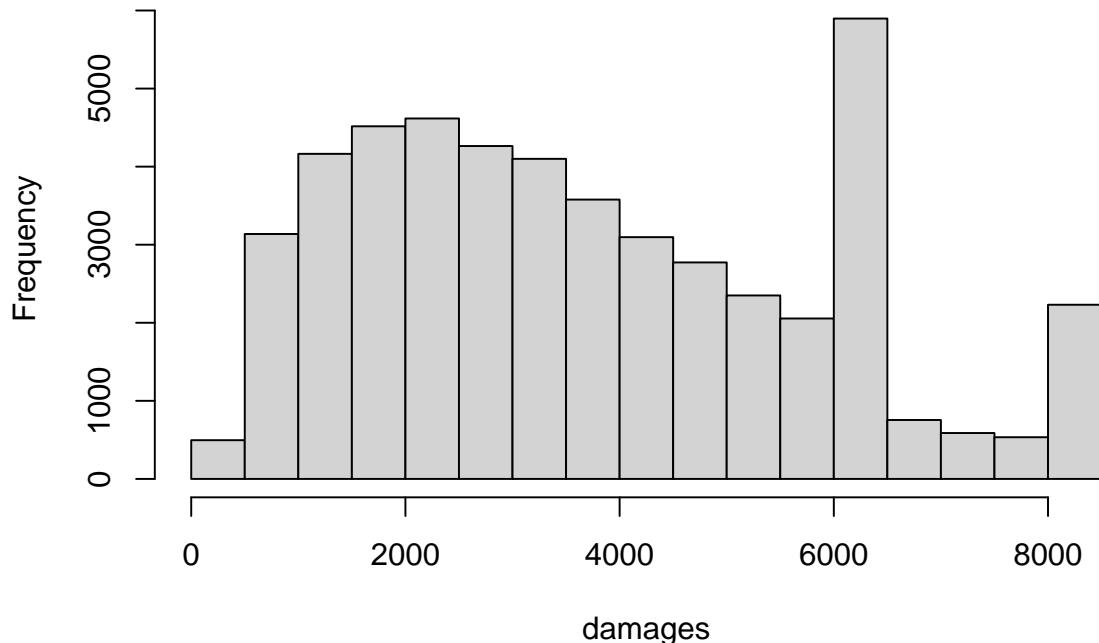
```
dataset <- readxl::read_xlsx("Practice_Final_Datasets/practice_q1_data.xlsx")  
  
hist(dataset$claim, main = "claims after application of deductibles and limits")
```

claims after application of deductibles and limits



```
damages <- dataset$claim + dataset$deductible  
hist(damages, main = "censored damage values before application of the deductible")
```

censored damage values before application of the deductible



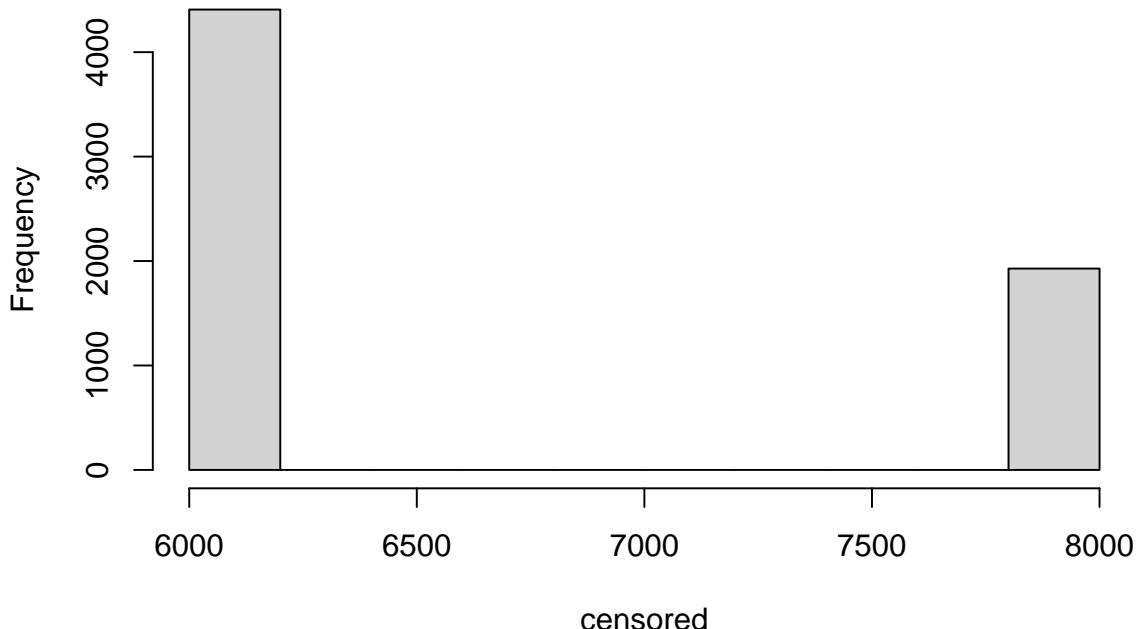
In both histograms there are two large spikes at around 6000 and 8000, which are typical of an aggregation of claim amounts due to the application of policy limits. The presence of deductibles is more subtle, and most obvious in the second histogram: there seems to be less claims in the first 500 than one would expect when looking at the shape of the distribution otherwise.

- (b) Provide a histogram of those claims which have reached their limit. What are all the possible amounts for limits in this dataset? What is the number of claims that had reached the highest of those limits? (1 mark)

The histogram is obtained by filtering for those claims which are equal to their limit. The possible amounts are 6000 and 8000 (use, for instance, the R function `unique` on the filtered claims), and the number of censored claims at 8000 is 1928.

```
censored <- dataset$claim[dataset$claim == dataset$limit]
hist(censored)
```

Histogram of censored



```
unique(censored)
```

```
## [1] 6000 8000
```

```
length(censored[censored == 8000])
```

```
## [1] 1928
```

- (c) We want to fit a distribution that describes the raw damage cost. For the 1st and 21st observations, write an expression for their contribution to the overall likelihood of this dataset, in terms of the pdf f , cdf F , and the relevant numerical values in the dataset (for instance, something like " $f(100) \times F(200)$ "). (2 marks)

Let's extract the 1st and 21st claims

```
# first claim:  
dataset[1, ]
```

```
## # A tibble: 1 x 3  
##   claim limit deductible  
##   <dbl> <dbl>      <dbl>  
## 1  2297  6000       500
```

```
# 21st claim:  
dataset[21, ]
```

```

## # A tibble: 1 x 3
##   claim limit deductible
##   <dbl>    <dbl>      <dbl>
## 1 6000    6000       250

```

It turns out that the second claim is censored, but not the first, which will require a different treatment. Furthermore, one should add back the deductible to the claim, as we are modelling the damage variable. In the end we get:

- likelihood of the first claim: $f(2297 + 500)/(1 - F(500))$
 - likelihood of the 21st claim: $(1 - F(6000 + 250))/(1 - F(250))$ and the contribution to the overall likelihood is $f(2297 + 500)/(1 - F(500)) \times (1 - F(6000 + 250))/(1 - F(250))$
- (d) Ignoring truncation and censoring, fit a gamma distribution to the data using the method of moments. Provide the corresponding numerical estimates for the `shape` and `rate` parameters. (1 mark)

This is easily done using `fitdist` function or the sample mean and variance of the damages (note that the mme technique matches the population variance instead of sample variance, both approaches are acceptable):

```

gamma.mme = fitdistrplus::fitdist(damages, "gamma", method = "mme",
                                    order = 1:2)
gamma.mme$estimate

```

```

##           shape          rate
## 3.1456555180 0.0008381131

n <- length(damages)
shape0 <- mean(damages)^2/var(damages)
rate0 <- mean(damages)/var(damages)
shape0

## [1] 3.145592

rate0

## [1] 0.000838096

```

So that we have `shape`=3.1456555 and `rate`=8.3811306 $\times 10^{-4}$.

- (e) Calculate the overall negative loglikelihood of the whole dataset, if one assumes a gamma distribution with parameters `shape`=2, and `rate`=0.0005. Include the R code used for this specific part (and not earlier parts (a) to (d)). You are not required to optimise this negative loglikelihood, but only to calculate its value evaluated at `shape`=2, and `rate`=0.0005. (4 marks)

One can write the negative loglikelihood from scratch, or simply copy paste the function provided in the lecture slides:

```

negloglik <- function(pdf, cdf, param, x, deduct, limitI) {
  # Function returns the negative log likelihood of the censored
  # and truncated dataset. Each data point's contribution to the
  # log likelihood depends on the theoretical distribution pdf and
  # cdf and also the deductible and limit values to adjust for
  # truncation and censoring
  PL <- do.call(cdf, c(list(q = deduct), param))
  PX <- do.call(cdf, c(list(q = x), param))
  fX <- do.call(pdf, c(list(x = x), param))
  lik.contr <- ifelse(limitI, log(1 - PX), log(fX)) - log(1 - PL)
  return(-sum(lik.contr))
}

answer <- negloglik(dgamma, pgamma, param = list(shape = 2,
  rate = 5e-04), damages, dataset$deductible, dataset$claim ==
  dataset$limit)

```

so that the answer is 394047.4.

Question 2

You are told that the following function

$$\phi_\theta(t) = e^{-t^{\frac{1}{\theta}}}, \quad \theta \in [1, \infty)$$

is either the generator function or the inverse generator function of an Archimedean copula.

- (a) By examining the relevant property of an Archimedean copula, determine whether the function is a generator function or an inverse generator function. (1 mark)

The relevant property is $\psi(1) = 0$ if ψ is a generator function. Here $\phi_\theta(1) = e^{-1} \neq 0$, hence ϕ_θ is an inverse generator function $\phi_\theta \equiv \psi_\theta^{-1}$.

- (b) By finding the inverse of $\phi_\theta(t)$ and following from part (a), express the corresponding bivariate Archimedean copula defined as

$$C_\theta(u, v) = \psi_\theta^{-1}(\psi_\theta(u) + \psi_\theta(v)).$$

in terms of u and v . (2 marks)

$$\begin{aligned} \phi_\theta^{-1}(t) &\equiv \psi_\theta(t) = (-\log t)^\theta \\ \Rightarrow C_\theta(u, v) &= \phi_\theta(\phi_\theta^{-1}(u) + \phi_\theta^{-1}(v)) \\ &= e^{-((-\log u)^\theta + (-\log v)^\theta)^{\frac{1}{\theta}}} \end{aligned}$$

- (c) You are also told that the copula function is an increasing function of $\theta \in [1, \infty)$. Show that the copula does not contain the countermonotonicity copula. (2 marks)

For $\theta = 1$, we obtain

$$C_1(u, v) = e^{-((-\log u) + (-\log v))} = uv$$

Since $(u - 1)(v - 1) > 0 \Rightarrow uv > u + v - 1$ for $0 < u < 1, 0 < v < 1$, the fact that the copula is an increasing function of θ means that the independence copula uv is its lower bound, and so the copula does not contain the Frechet lower bound of $\max(u + v - 1, 0)$ defined by the countermonotonicity copula.

Question 3

The attached dataset `practice_q3_data.xlsx` corresponds to the `hurricanehist` dataset in the R package `CASdatasets`. It is described in the package documentation as “The dataset consists of 2010 observations for all tropical cyclones in the NHC best track record over the period 1899-2006. Each observation contains per cyclone maximum wind speeds and other relevant information.”

The vignette goes on with describing: “`hurricanehist` contains 7 columns:

- `Year`: The Year.
- `Region`: The region among “Basin”, “East Florida”, “Gulf”, “US”.
- `Windmax`: The maximum windspeed in knot (1kt = 0.51 m/s).
- `NAO`: the North Atlantic Oscillation (NAO) index as an indicator of storm steering.
- `SOI`: the Southern Oscillation Index (SOI) as an indicator of El Nino-Southern Oscillation.
- `SST`: the Atlantic sea-surface temperature (SST) as an indicator of cyclone energy.
- `SSTmda`: the SST mda.”

The original source of the data is http://myweb.fsu.edu/jelsner/_site/.

Perform the following tasks, which are focused on the “US” region only:

- (a) Consider the maximum windspeeds in the region “US” only. Transform the data into a time series of yearly maxima, and include a time series plot (with the correct frequency and start date) in your answer, as well as the first 10 values. (3 marks)

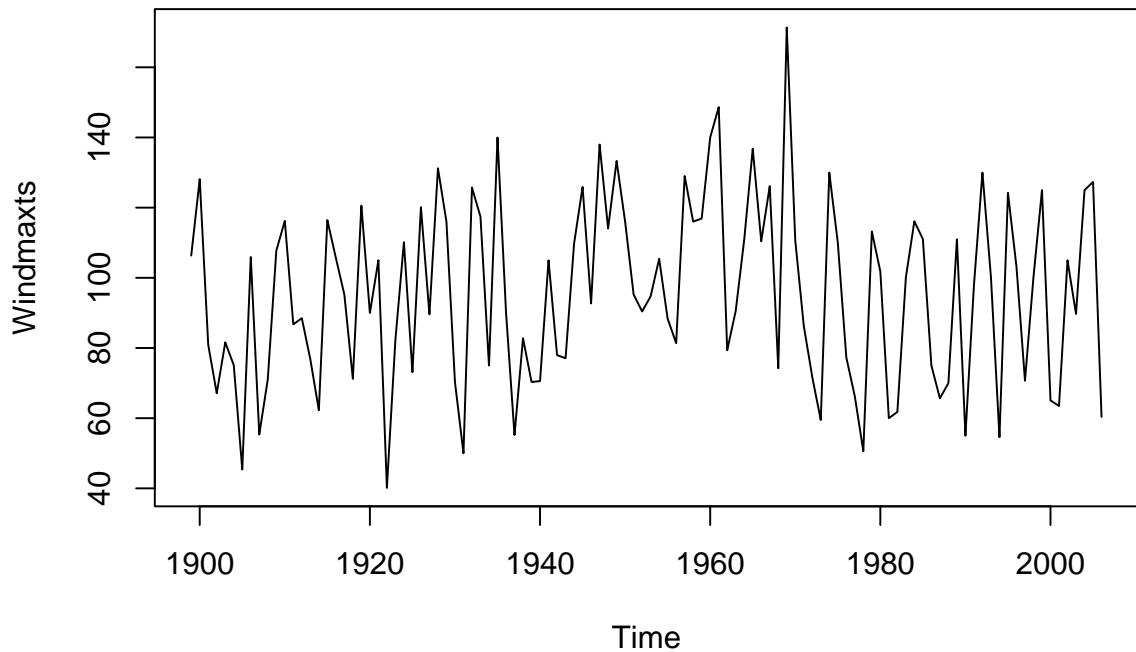
```
dataset <- readxl::read_xlsx("Practice_Final_Datasets/practice_q3_data.xlsx")

library(dplyr)

USmax <- dataset %>%
  filter(dataset$Region == "US") %>%
  group_by(Year) %>%
  summarise(Max_Windmax = max(Windmax))

Windmaxts <- ts(data = USmax$Max_Windmax, start = min(USmax$Year),
  frequency = 1)

plot(Windmaxts)
```



```
USmax$Max_Windmax[1:10]
```

```
## [1] 106.35318 128.12384 80.88609 67.09340 81.62150 75.12913 45.38930
## [8] 105.94556 55.35019 71.06642
```

- (b) Fit an appropriate Generalised Extreme Value (GEV) distribution for the ***yearly maxima*** of windspeed in the “US” region. Report the numerical values (and their standard errors) of the distribution parameters, and include the relevant diagnostic plots. State the type of the fitted GEV distribution and state the range of values of the yearly maxima. Discuss the goodness of fit of your results based on the diagnostic plots. (5 marks)

Because we already have a dataset of yearly maxima, there is no need to use the block maxima method. If we wanted a distribution of maxima per event, then we would need to use the block maxima method.

```
fit.USmax <- extRemes::fevd(USmax$Max_Windmax)

fit.USmax

##
## extRemes::fevd(x = USmax$Max_Windmax)
##
## [1] "Estimation Method used: MLE"
```

```

## 
## 
## Negative Log-Likelihood Value: 505.7602
## 
## 
## Estimated parameters:
##   location      scale      shape
## 85.4656711 25.4042988 -0.2275732
## 
## Standard Error Estimates:
##   location      scale      shape
## 2.68824411 1.91816436 0.05739781
## 
## Estimated parameter covariance matrix.
##           location      scale      shape
## location  7.22665640 0.48676130 -0.056153148
## scale     0.48676130 3.67935453 -0.060548742
## shape    -0.05615315 -0.06054874  0.003294509
## 
## AIC = 1017.52
## 
## BIC = 1025.567

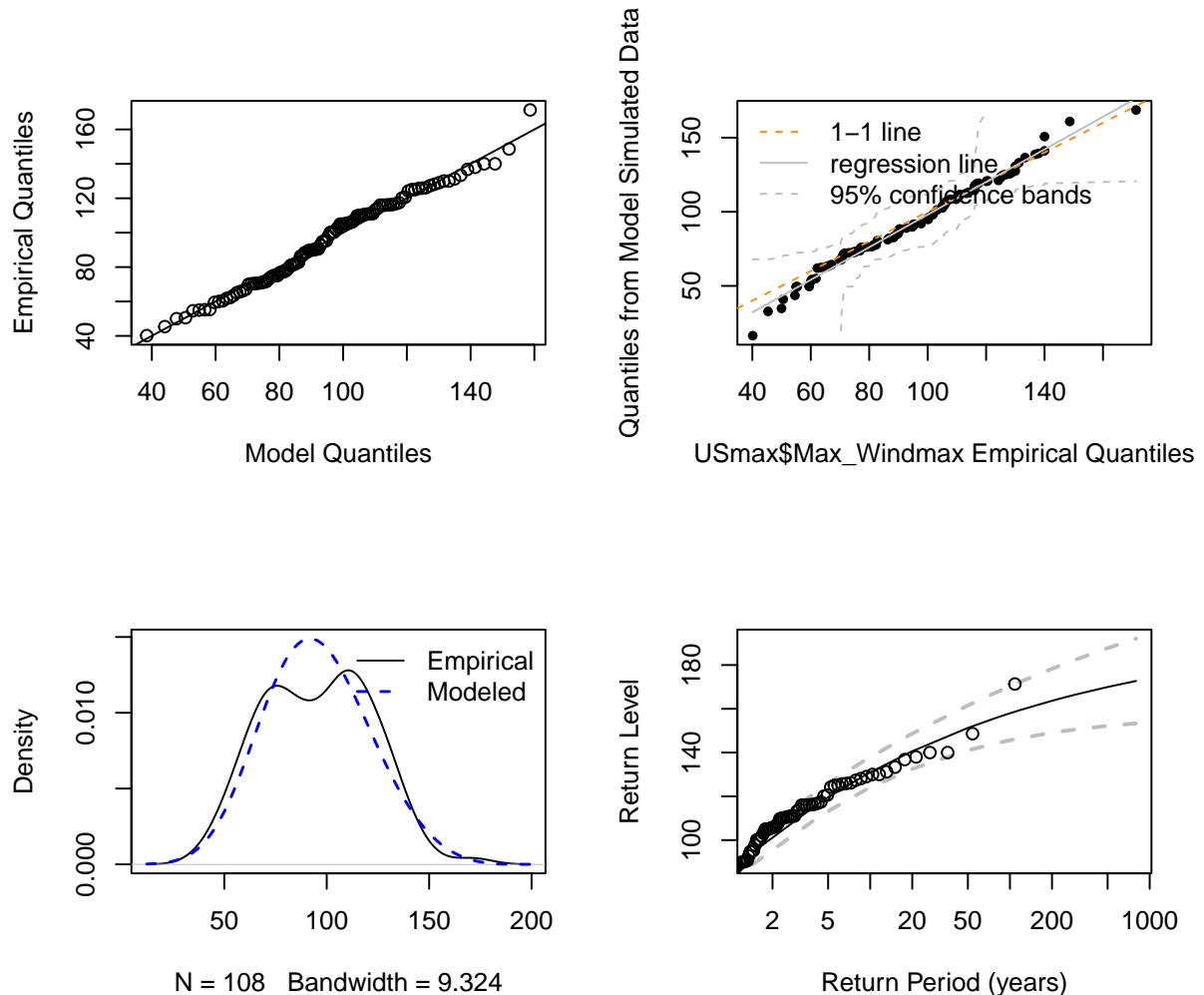
```

The estimated parameters of location, scale and shape and their standard errors are given above.

Since the shape parameters are negative and significant, it belongs to the upper bounded Weibull type with a support of $x \leq \mu - \frac{\sigma}{\gamma} = 197.097$.

```
plot(fit.USmax)
```

```
extRemes::fevd(x = USmax$Max_Windmax)
```



The diagnostic plots look reasonably good:

- the QQ plot (data vs model) is good, with most dots along the diagonal.
- the QQ plot (data vs simulated from model) is also quite good, with all dots well within the confidence bounds, and the regression line almost matching the diagonal.
- the density plot is well centered, and the bimodality of the data is simply smoothed into the unimodal EVD.

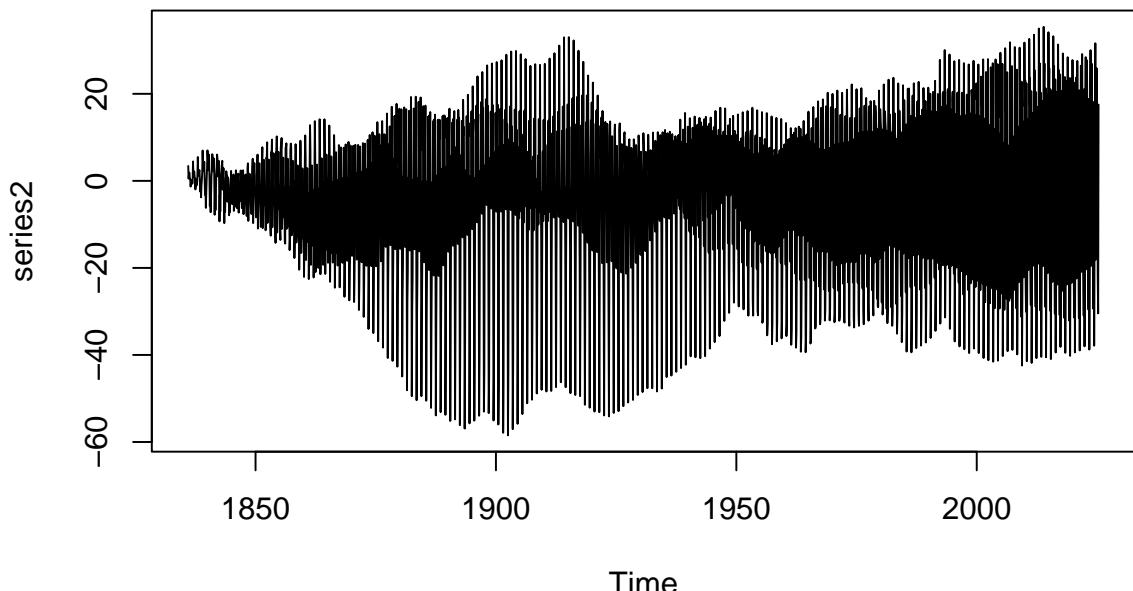
Question 4

The dataset in `practice_q4_data.xlsx` contains monthly time series x_t starting from January 1836.

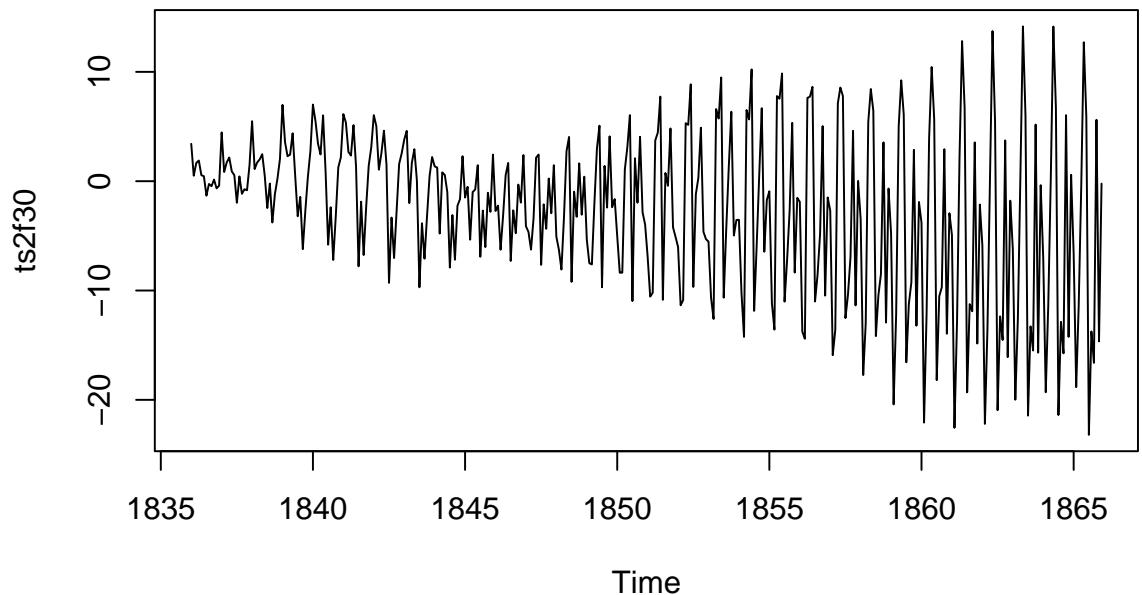
- (a) Explore the time series x_t using relevant plots and discuss whether the series is likely to be stationary. Hint: you may want to plot the series for a specific sub-period and/or specify the maximum lag of your plot for better visualisation. (4 marks)

```
ts2 <- ts(data = read_xlsx("Practice_Final_Datasets/practice_q4_data.xlsx"),
           start = 1836, frequency = 12)

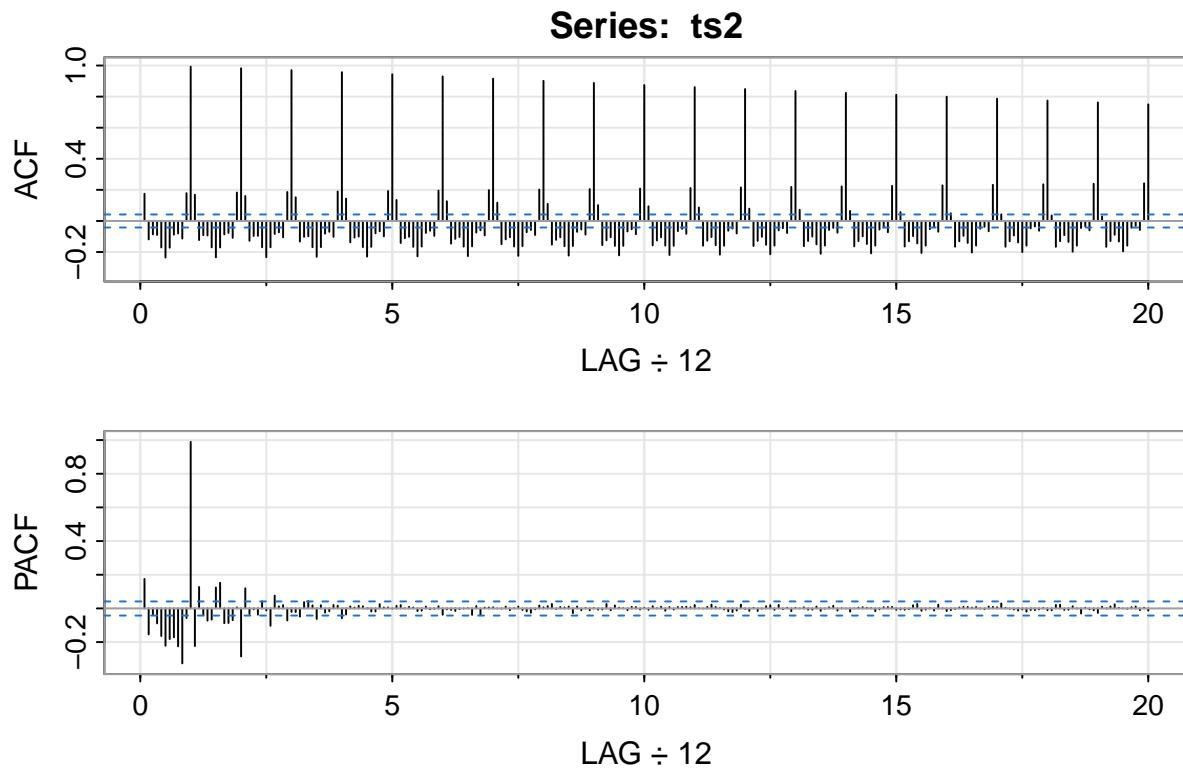
plot(ts2)
```



```
# plot the first 30 years of monthly data
ts2f30 <- ts(data = ts2[1:360], start = 1836, frequency = 12)
plot(ts2f30)
```



```
# plot the acf and pacf for up to a maximum lag of 240  
acf2(ts2, max.lag = 240)
```



There are obvious patterns of (i) non-constant mean, (ii) seasonal / cyclical component and (iii) increasing variance over time, suggesting that the series is not stationary.

We choose to plot the ACF and PACF up to a maximum lags of 20 years (240 months).

The extremely slow decaying speed of ACF at lag 12h also indicates that the series is not stationary.

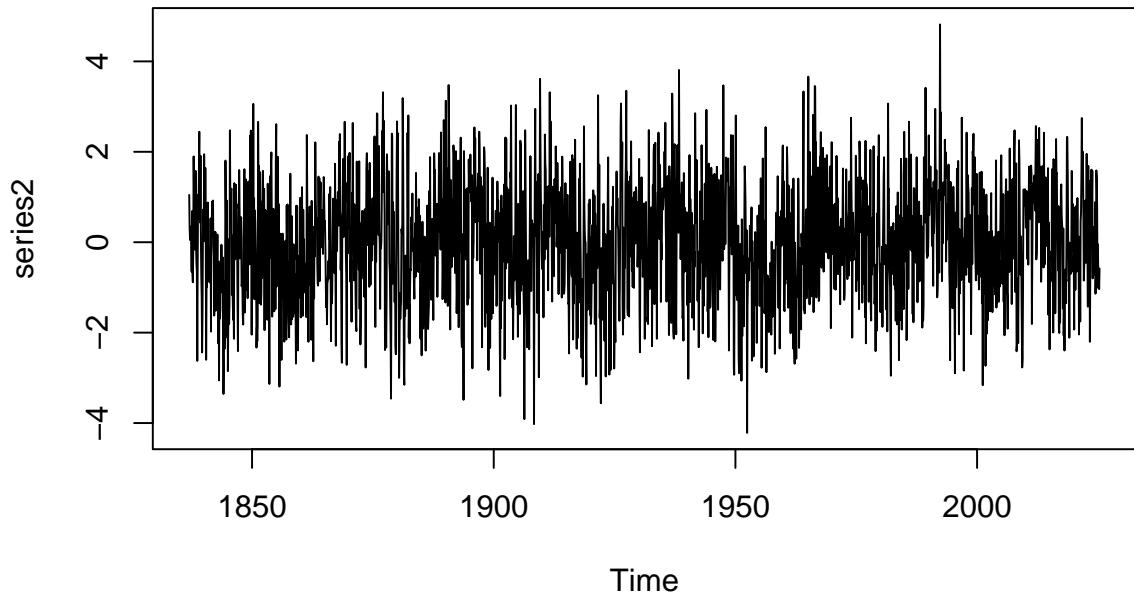
- (b) Using the plot in part (a), explain why de-trending via regression may not be a good idea for the series. (1 mark)

Looking at the data over close to 200 years, there is no obvious long-term deterministic trend underlying the data, due to the presence of seasonal effect.

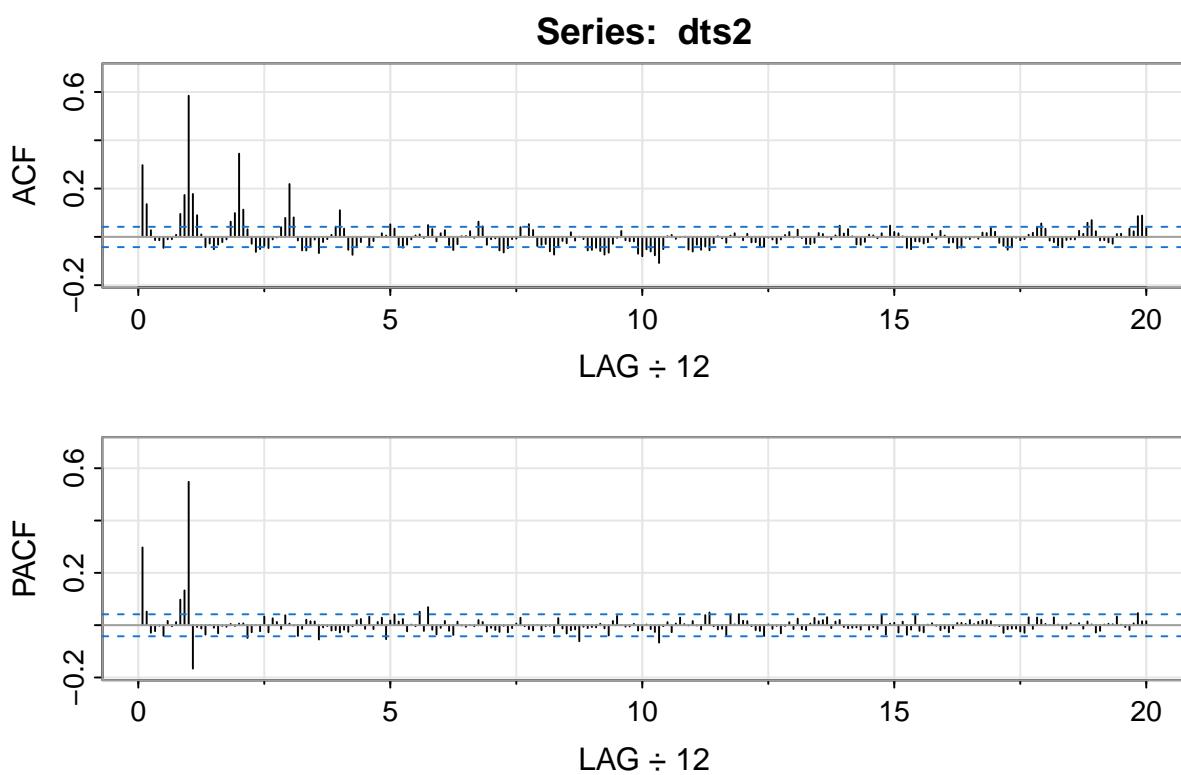
- (c) Perform the necessary next step in the modelling by considering potential integrated orders of
- $d = 0, D = 1$
 - $d = 1, D = 0$

Comment on your findings, verify against your discussion in part (a) and determine the appropriate integrated orders. (4 marks)

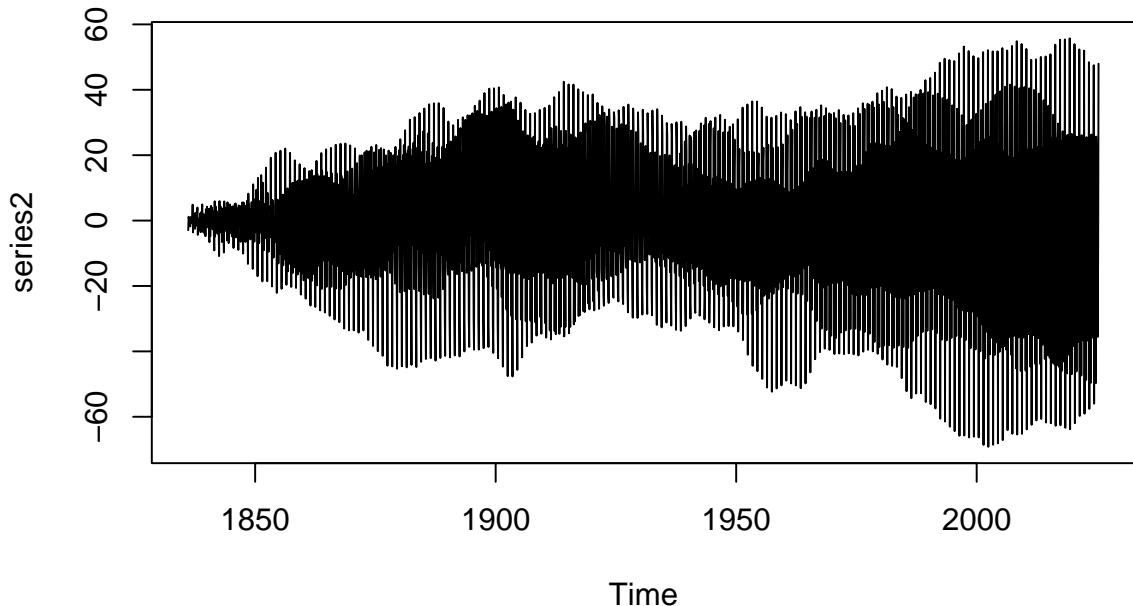
```
dts2 <- diff(ts2, lag = 12)
plot(dts2)
```



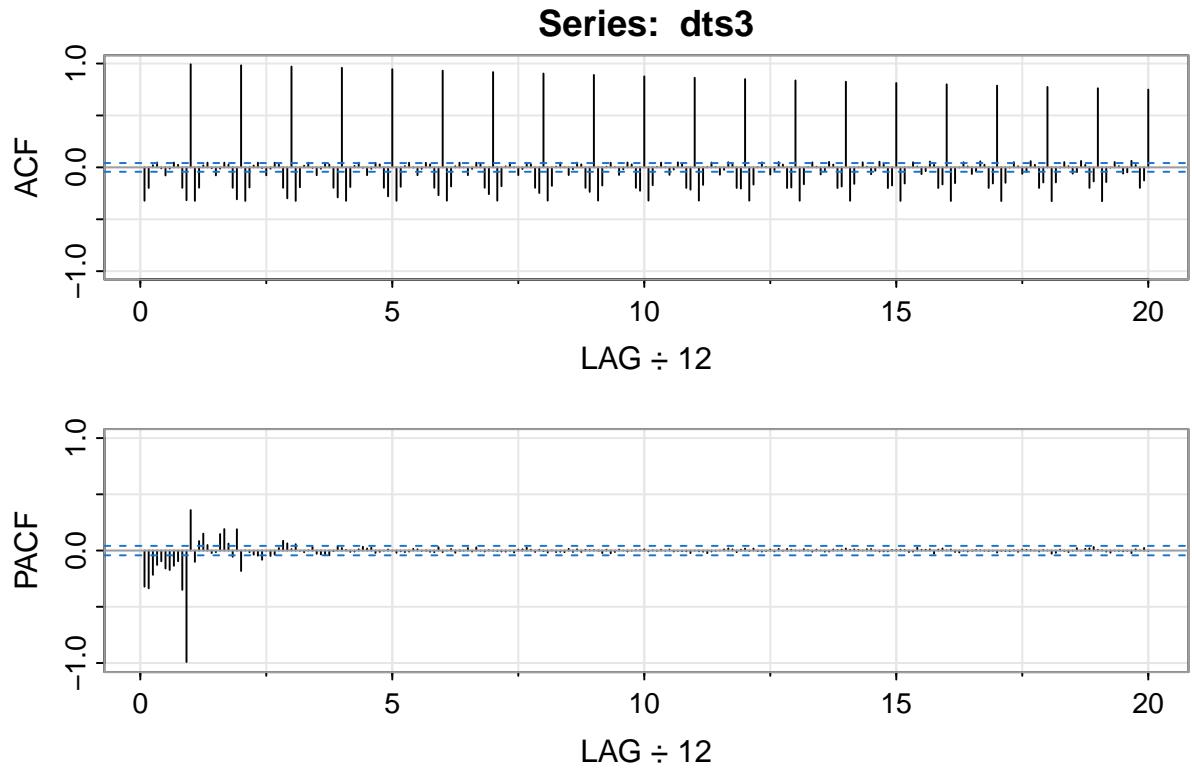
```
acf2(dts2, max.lag = 240)
```



```
dts3 <- diff(ts2, lag = 1)
plot(dts3)
```



```
acf2(dts3, max.lag = 240)
```



The plot of the seasonally-differenced series ($D = 1$ with $s = 12$) appears to be stationary and the ACF at lags $12h$ appears to be tailing off, a marked improvement from the very slow decaying speed at lag $12h$ in the original series as shown in part (a).

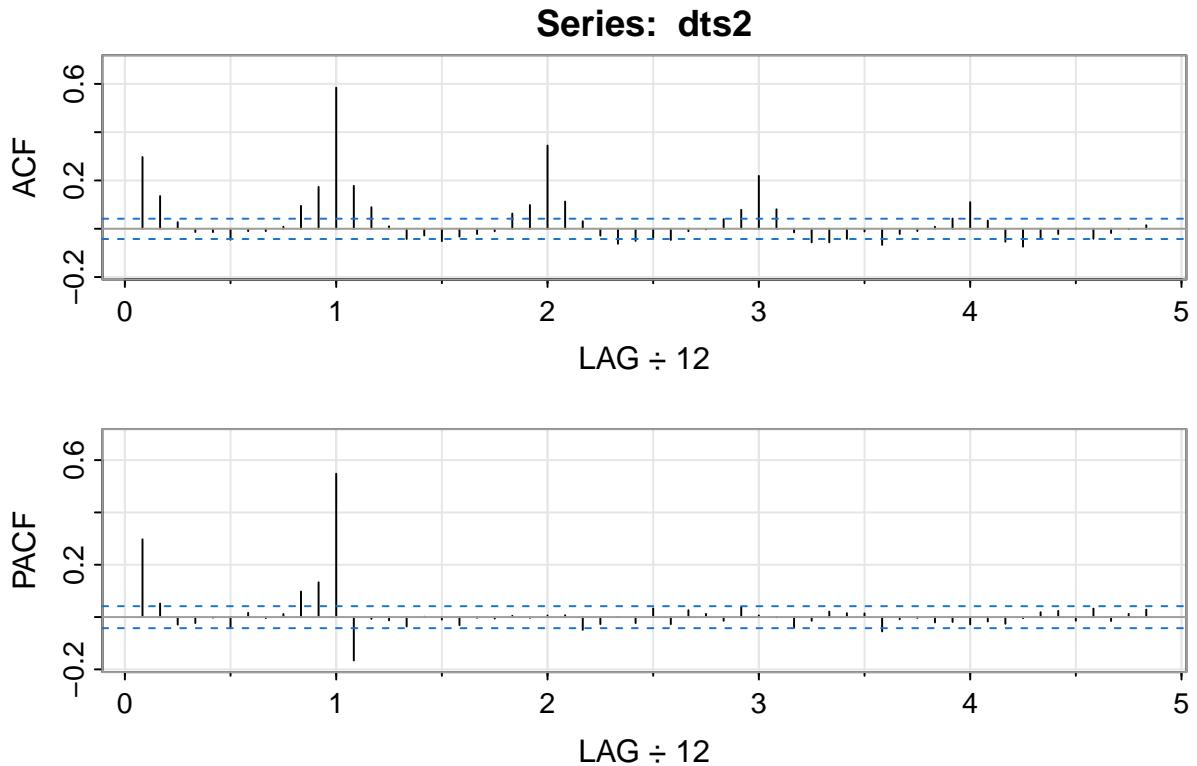
For the case of $d = 1$, the non-stationarity problem due to slow decaying of ACF remains and the plot of the non-seasonal-differenced series are similar to that of the original series.

Hence we select $d = 0, D = 1, s = 12$ to produce the desired stationary manipulated series.

- (d) Using the ACF and PACF plots of the chosen manipulated series from part (c), identify at least three candidate time series models and discuss the rationales for these choices. (3 marks)

Zooming in the plot for a lag of up to 5 years

```
acf2(dts2)
```



The PACF cuts off at lag 12 and the ACF tails off at lag $12h$, indicating a $sAR(1)_{12}$ model with $P = 1, Q = 0, s = 12$.

At non-seasonal lags, the ACF appear to tail off on either side of the seasonal lags and the ACF appears to cut off at a lag of 1 on either side of lags 0 and 12, suggesting $ARMA(1, 0)$ with $p = 1, q = 0$. So the candidate model is $ARMA(1, 0) \times sAR(1)_{12}$ for the seasonally-differenced series.

We also consider another three candidate models of

- $ARMA(0, 2) \times sAR(1)_{12}$
- $ARMA(2, 0) \times sAR(1)_{12}$
- $ARMA(1, 2) \times sAR(1)_{12}$

since the patterns at the non-seasonal lags are numerically dependent on the relevant seasonal components and other time series models could lead to the same observed patterns.

- (e) By performing model estimations for the candidate models specified in part (d) and inspecting the resulting diagnostic plots, verify against your discussions in part (d) and justify the final selected model for the original series given in the dataset. Include the relevant parameters in the model and express the series x_t in the estimated model equation. (6 marks)

```
dts2.ar1 <- sarima(dts2, 1, 0, 0, 1, 0, 0, 12)
```

```
## initial value 0.255972
```

```

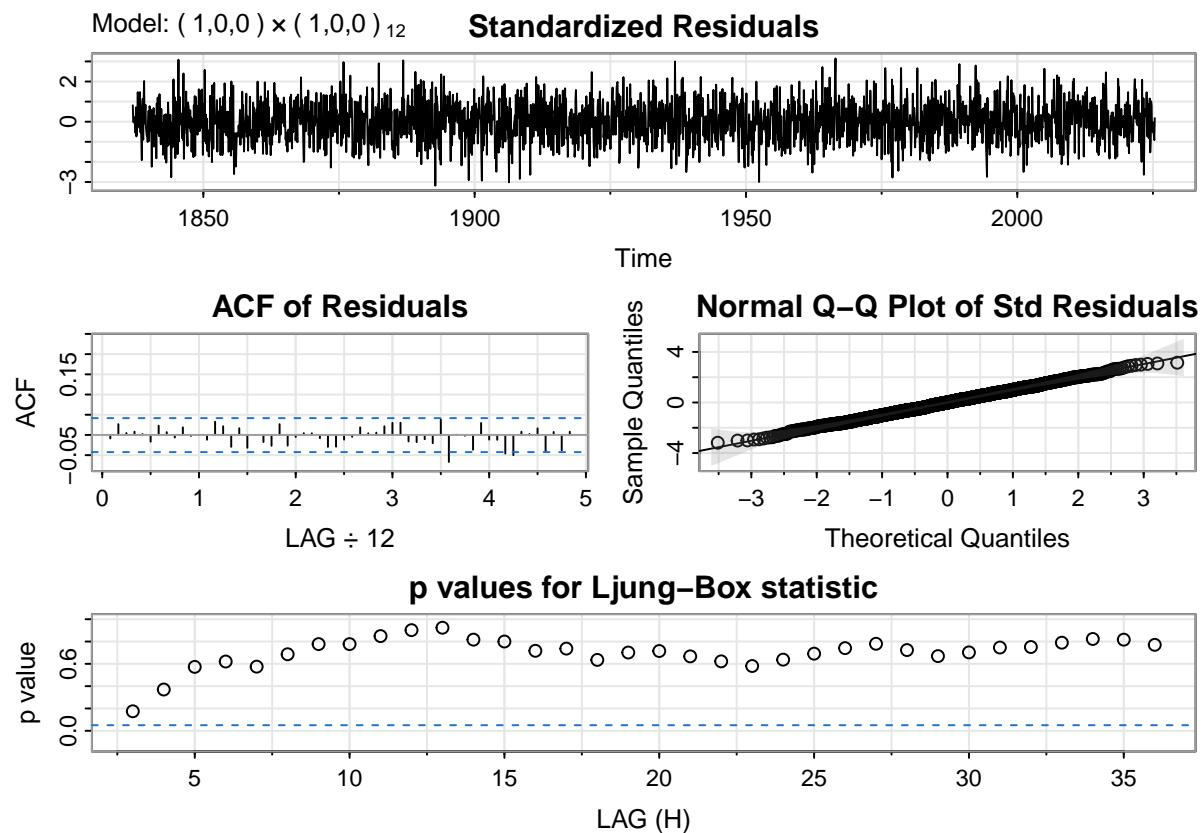
## iter  2 value 0.000712
## iter  3 value 0.000707
## iter  4 value 0.000707
## iter  5 value 0.000707
## iter  6 value 0.000707
## iter  7 value 0.000707
## iter  8 value 0.000707
## iter  9 value 0.000707
## iter 10 value 0.000707
## iter 11 value 0.000707
## iter 12 value 0.000707
## iter 13 value 0.000707
## iter 14 value 0.000707
## iter 15 value 0.000706
## iter 16 value 0.000706
## iter 17 value 0.000706
## iter 17 value 0.000706
## final  value 0.000706
## converged
## initial  value -0.000029
## iter  2 value -0.000031
## iter  3 value -0.000031
## iter  4 value -0.000031
## iter  5 value -0.000032
## iter  6 value -0.000032
## iter  7 value -0.000032
## iter  8 value -0.000032
## iter  9 value -0.000032
## iter 10 value -0.000032
## iter 11 value -0.000032
## iter 12 value -0.000032
## iter 13 value -0.000032
## iter 14 value -0.000032
## iter 15 value -0.000032
## iter 16 value -0.000032
## iter 17 value -0.000032
## iter 17 value -0.000032
## final  value -0.000032
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##      Estimate     SE t.value p.value

```

```

## ar1      0.2940 0.0201 14.6344  0.0000
## sar1     0.5830 0.0170 34.2889  0.0000
## xmean   -0.0018 0.0708 -0.0261  0.9792
##
## sigma^2 estimated as 0.9976934 on 2258 degrees of freedom
##
## AIC = 2.841351  AICc = 2.841356  BIC = 2.851477
##

```



Both AR and sAR coefficients are significant. The ACF of residuals and the p-values of Ljung-Box statistic suggest white noise residuals. The constant term is not significant.

```
dts2.ar2 <- sarima(dts2, 2, 0, 0, 1, 0, 0, 12)
```

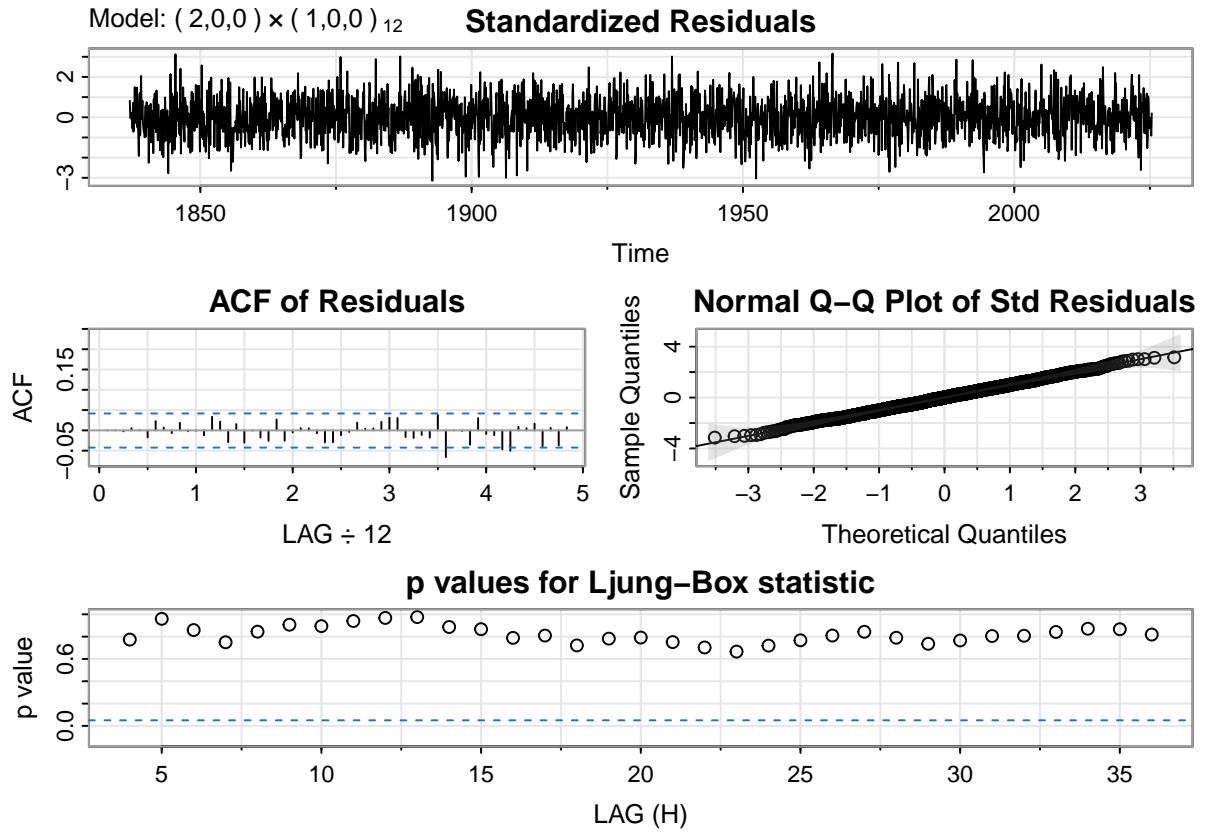
```

## initial  value 0.256184
## iter    2 value 0.007167
## iter    3 value 0.000902
## iter    4 value 0.000527
## iter    5 value 0.000502
## iter    6 value 0.000500
## iter    7 value 0.000500
## iter    8 value 0.000499
## iter    9 value 0.000499

```

```

## iter   10 value 0.000499
## iter   11 value 0.000499
## iter   12 value 0.000499
## iter   13 value 0.000499
## iter   14 value 0.000499
## iter   15 value 0.000499
## iter   16 value 0.000499
## iter   17 value 0.000499
## iter   18 value 0.000499
## iter   19 value 0.000499
## iter   20 value 0.000499
## iter   21 value 0.000499
## iter   21 value 0.000499
## final  value 0.000499
## converged
## initial value -0.000443
## iter    2 value -0.000445
## iter    3 value -0.000445
## iter    4 value -0.000445
## iter    5 value -0.000446
## iter    6 value -0.000446
## iter    7 value -0.000446
## iter    8 value -0.000446
## iter    9 value -0.000446
## iter   10 value -0.000446
## iter   10 value -0.000446
## final  value -0.000446
## converged
## <><><><><><><><><><><><><><>
## 
## Coefficients:
##             Estimate      SE t.value p.value
## ar1        0.2855  0.0210 13.5896  0.0000
## ar2        0.0288  0.0210  1.3687  0.1712
## sar1       0.5823  0.0170 34.2049  0.0000
## xmean     -0.0018  0.0728 -0.0254  0.9797
## 
## sigma^2 estimated as 0.9968736 on 2257 degrees of freedom
## 
## AIC = 2.841407  AICc = 2.841415  BIC = 2.854064
## 
```



Compared to AR(1) model, the second AR coefficient is not significant, so we drop this model.

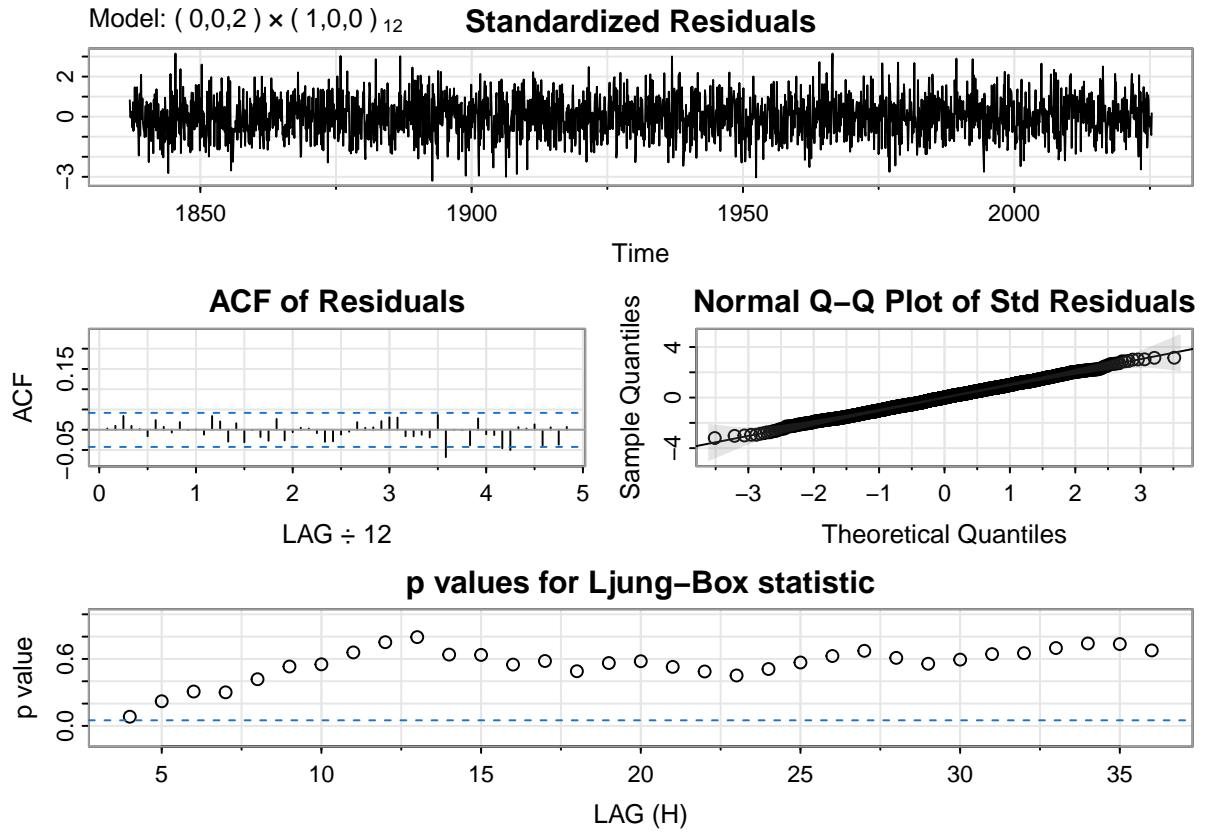
```
dts2.ma2 <- sarima(dts2, 0, 0, 2, 1, 0, 0, 12)
```

```
## initial value 0.255889
## iter 2 value 0.001432
## iter 3 value 0.000785
## iter 4 value 0.000748
## iter 5 value 0.000747
## iter 6 value 0.000747
## iter 7 value 0.000747
## iter 8 value 0.000747
## iter 9 value 0.000747
## iter 10 value 0.000747
## iter 11 value 0.000747
## iter 12 value 0.000747
## iter 13 value 0.000747
## iter 14 value 0.000747
## iter 15 value 0.000747
## iter 16 value 0.000747
## iter 17 value 0.000747
```

```

## iter 18 value 0.000747
## iter 19 value 0.000747
## iter 20 value 0.000747
## iter 20 value 0.000747
## iter 20 value 0.000747
## final value 0.000747
## converged
## initial value 0.000191
## iter 2 value 0.000188
## iter 3 value 0.000188
## iter 4 value 0.000188
## iter 5 value 0.000188
## iter 6 value 0.000188
## iter 7 value 0.000188
## iter 8 value 0.000187
## iter 9 value 0.000187
## iter 10 value 0.000187
## iter 11 value 0.000187
## iter 12 value 0.000187
## iter 13 value 0.000187
## iter 14 value 0.000187
## iter 15 value 0.000187
## iter 16 value 0.000187
## iter 17 value 0.000187
## iter 18 value 0.000187
## iter 19 value 0.000187
## iter 20 value 0.000187
## iter 20 value 0.000187
## final value 0.000187
## converged
## <><><><><><><><><><><><>
## 
## Coefficients:
##             Estimate      SE t.value p.value
## ma1       0.2832 0.0208 13.5951 0.0000
## ma2       0.1005 0.0205  4.9022 0.0000
## sar1      0.5820 0.0170 34.1591 0.0000
## xmean    -0.0018 0.0690 -0.0266 0.9788
## 
## sigma^2 estimated as 0.9981416 on 2257 degrees of freedom
## 
## AIC = 2.842674  AICc = 2.842682  BIC = 2.855331
##

```



The MA and sAR coefficients are significant. The ACF of residuals looks good but one of the p-values of the Ljung–Box statistic do not suggest white noise residuals. The constant term is not significant.

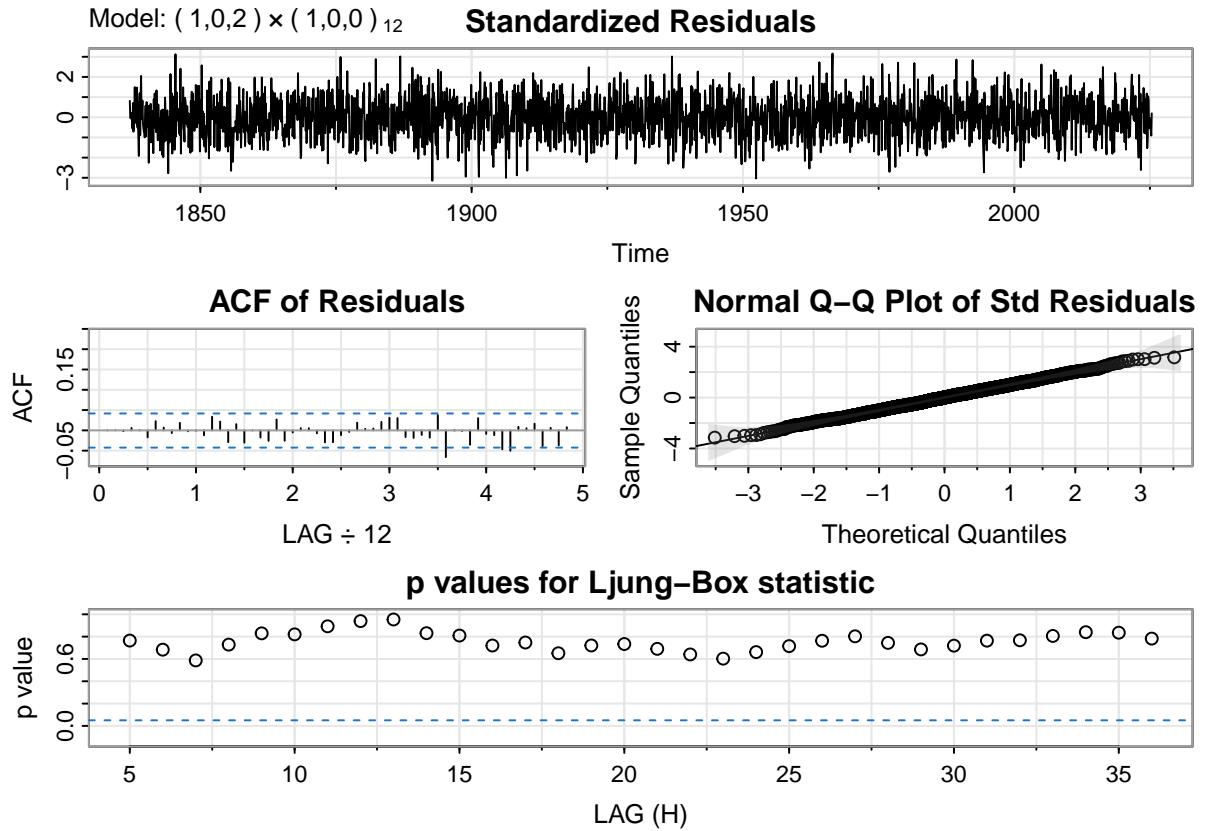
```
dts2.ar1ma2 <- sarima(dts2, 1, 0, 2, 1, 0, 0, 12)
```

```
## initial value 0.255972
## iter 2 value 0.047151
## iter 3 value 0.005813
## iter 4 value 0.001753
## iter 5 value 0.000579
## iter 6 value 0.000561
## iter 7 value 0.000559
## iter 8 value 0.000527
## iter 9 value 0.000476
## iter 10 value 0.000388
## iter 11 value 0.000318
## iter 12 value 0.000282
## iter 13 value 0.000280
## iter 14 value 0.000279
## iter 15 value 0.000279
## iter 16 value 0.000279
```

```

## iter 17 value 0.000279
## iter 18 value 0.000279
## iter 19 value 0.000279
## iter 20 value 0.000279
## iter 21 value 0.000279
## iter 22 value 0.000279
## iter 23 value 0.000279
## iter 24 value 0.000279
## iter 25 value 0.000279
## iter 26 value 0.000279
## iter 27 value 0.000279
## iter 28 value 0.000279
## iter 28 value 0.000279
## iter 28 value 0.000279
## final value 0.000279
## converged
## initial value -0.000442
## iter 2 value -0.000444
## iter 3 value -0.000444
## iter 4 value -0.000444
## iter 5 value -0.000444
## iter 6 value -0.000445
## iter 7 value -0.000445
## iter 8 value -0.000445
## iter 9 value -0.000445
## iter 10 value -0.000445
## iter 11 value -0.000445
## iter 12 value -0.000445
## iter 12 value -0.000445
## final value -0.000445
## converged
## <><><><><><><><><><><><><>
## 
## Coefficients:
##             Estimate      SE t.value p.value
## ar1       0.3555 0.1786   1.9901  0.0467
## ma1      -0.0699 0.1800  -0.3885  0.6977
## ma2       0.0089 0.0563   0.1579  0.8746
## sar1      0.5823 0.0170 34.1696  0.0000
## xmean    -0.0019 0.0727  -0.0255  0.9797
## 
## sigma^2 estimated as 0.9968755 on 2256 degrees of freedom
## 
## AIC = 2.842294  AICc = 2.842305  BIC = 2.857482
##

```



The MA coefficients θ_1 and θ_2 are not significant and the constant term is not significant, so we drop this model.

We refit the two remaining candidate models by setting `no.constant=TRUE`.

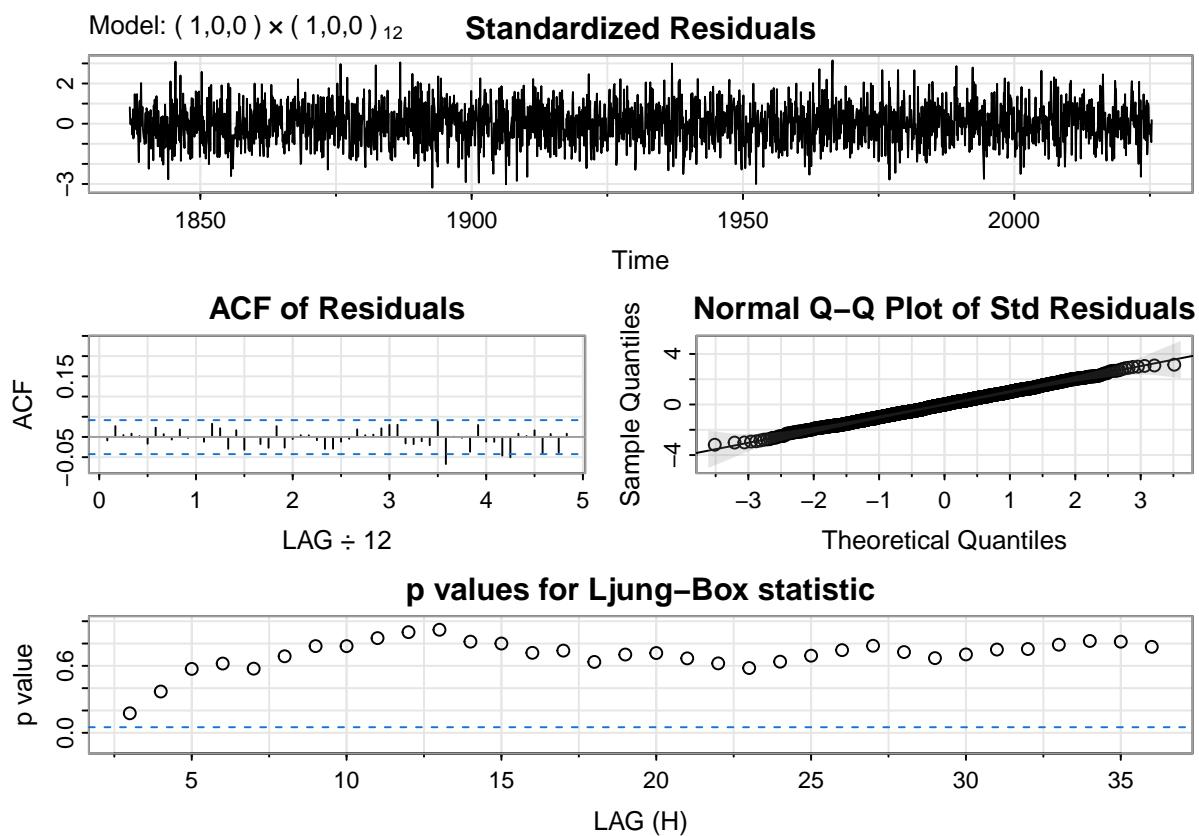
```
dts2.ar1 <- sarima(dts2, 1, 0, 0, 1, 0, 0, 12, no.constant = TRUE)
```

```
## initial value 0.255976
## iter 2 value 0.000714
## iter 3 value 0.000708
## iter 4 value 0.000708
## iter 5 value 0.000708
## iter 6 value 0.000708
## iter 6 value 0.000708
## final value 0.000708
## converged
## initial value -0.000029
## iter 2 value -0.000032
## iter 3 value -0.000032
## iter 4 value -0.000032
## iter 4 value -0.000032
## iter 4 value -0.000032
## final value -0.000032
```

```

## converged
## <><><><><><><><><><><><>
## 
## Coefficients:
##             Estimate      SE t.value p.value
## ar1       0.294 0.0201 14.6345      0
## sar1      0.583 0.0170 34.2890      0
## 
## sigma^2 estimated as 0.9976937 on 2259 degrees of freedom
## 
## AIC = 2.840467  AICc = 2.840469  BIC = 2.848061
##

```



```

dts2.ma2 <- sarima(dts2, 0, 0, 2, 1, 0, 0, 12, no.constant = TRUE)

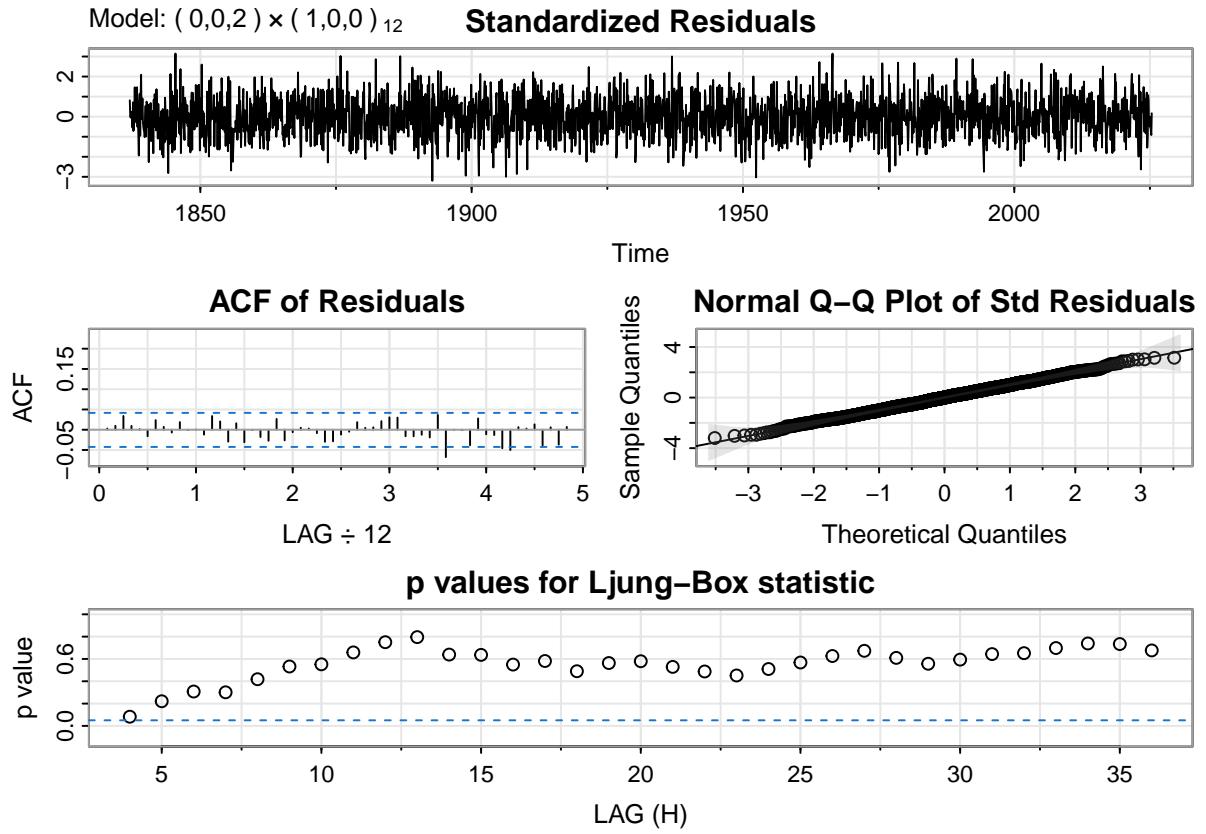
## initial value 0.255893
## iter 2 value 0.001433
## iter 3 value 0.000787
## iter 4 value 0.000749
## iter 5 value 0.000749
## iter 6 value 0.000748
## iter 7 value 0.000748

```

```

## iter    7 value 0.000748
## iter    7 value 0.000748
## final   value 0.000748
## converged
## initial  value 0.000190
## iter    2 value 0.000188
## iter    3 value 0.000187
## iter    4 value 0.000187
## iter    5 value 0.000187
## iter    5 value 0.000187
## iter    5 value 0.000187
## final   value 0.000187
## converged
## <><><><><><><><><><><><>
##
## Coefficients:
##             Estimate      SE t.value p.value
## ma1       0.2832  0.0208 13.5951      0
## ma2       0.1005  0.0205  4.9022      0
## sar1      0.5820  0.0170 34.1591      0
##
## sigma^2 estimated as 0.9981419 on 2258 degrees of freedom
##
## AIC = 2.84179  AICc = 2.841795  BIC = 2.851916
##

```



The diagnostics of both models are equally good. We select the first model $ARMA(1,0) \times sAR(1)_{12}$ because of the lower BIC, AIC and AICc in favour of model parsimony.

Note that we obtain the same fit below if we change the arguments of sarima accordingly.

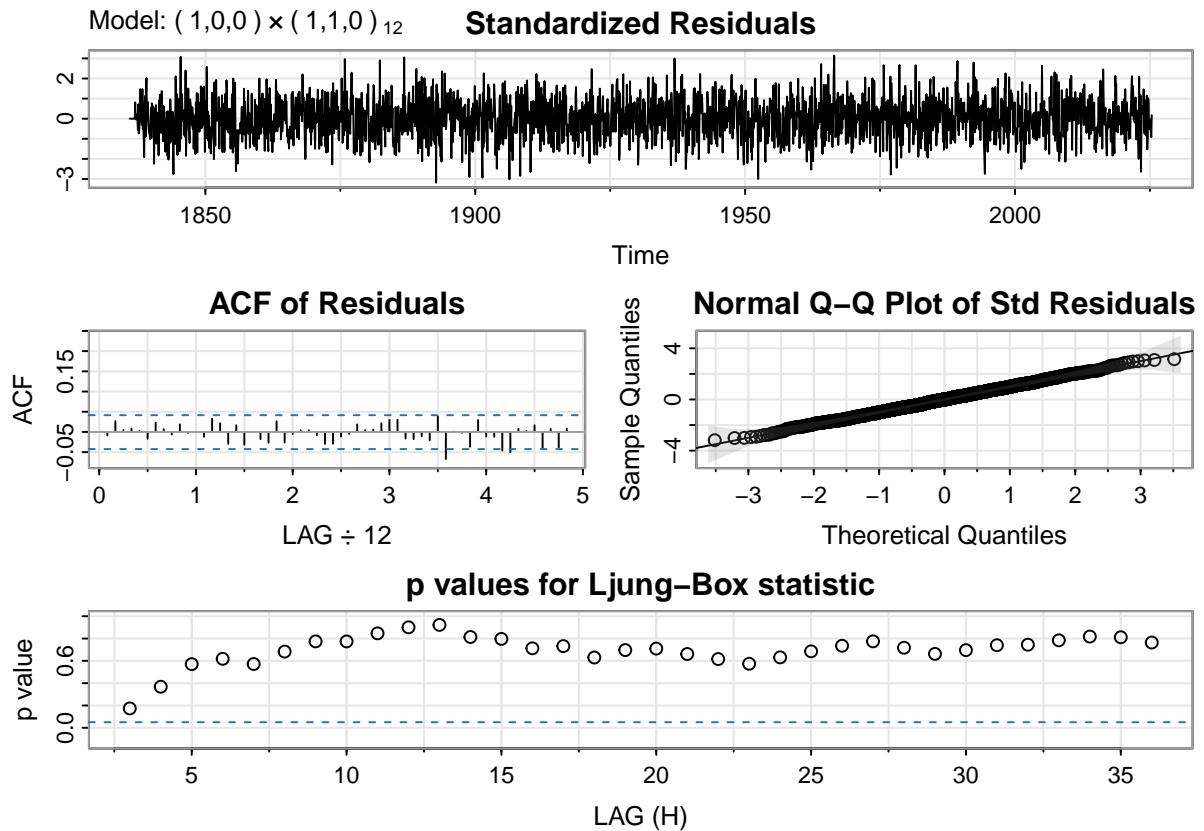
```
dts2.ar1b <- sarima(ts2, 1, 0, 0, 1, 1, 0, 12, no.constant = TRUE)
```

```
## initial value 0.255976
## iter 2 value 0.000714
## iter 3 value 0.000708
## iter 4 value 0.000708
## iter 5 value 0.000708
## iter 6 value 0.000708
## iter 6 value 0.000708
## final value 0.000708
## converged
## initial value -0.000029
## iter 2 value -0.000032
## iter 3 value -0.000032
## iter 4 value -0.000032
## iter 4 value -0.000032
## iter 4 value -0.000032
## final value -0.000032
```

```

## converged
## <><><><><><><><><><><><>
## 
## Coefficients:
##             Estimate      SE t.value p.value
## ar1       0.294 0.0201 14.6344      0
## sar1      0.583 0.0170 34.2890      0
## 
## sigma^2 estimated as 0.9976937 on 2259 degrees of freedom
## 
## AIC = 2.840467  AICc = 2.840469  BIC = 2.848061
##

```



The estimated model can be expressed as

$$(1 - 0.3258B)(1 - B^{12})(1 - 0.5823B^{12})x_t = w_t$$

$$(1 - 0.3258B - 0.5823B^{12} + 0.5155B^{13} + 0.5823B^{24} - 0.1897B^{25})x_t = w_t$$

$$\Rightarrow x_t = 0.3258x_{t-1} + 0.5823x_{t-12} - 0.5155x_{t-13} - 0.5823x_{t-24} + 0.1897x_{t-25} + w_t$$

with $\hat{\sigma}^2 = 0.9987$.

Question 5

Consider a time series x_t specified in the following model

$$x_t = -0.7x_{t-1} + 0.6x_{t-2} + w_t - 0.2w_{t-1}$$

where w_t is uncorrelated white noise with constant variance σ^2 .

- (a) Determine whether the model is invertible. (1 mark)

The MA polynomial is given by

$$\begin{aligned}\theta(B) &= 1 - 0.2B \\ \theta(z) &= 1 - 0.2z = 0 \\ z &= 5 > 1\end{aligned}$$

that is, the root is outside the unit circle, so the model is invertible.

- (b) Determine whether the model is causal. (2 marks)

The AR polynomial is given by

$$\begin{aligned}\phi(B) &= 1 + 0.7B - 0.6B^2 \\ \phi(z) &= 1 + 0.7z - 0.6z^2 = (1 + 1.2z)(1 - 0.5z) = 0 \\ |z| &= \left| -\frac{5}{6} \right| < 1 \text{ and } z = 2 > 1\end{aligned}$$

that is, one of the roots is inside the unit circle, so the model is not causal.

- (c) Simulate 300 values of the time series using a suitable *R* function and verify against your answer in part (b). (2 marks)

```
x <- arima.sim(list(order = c(2, 0, 1), ar = c(-0.7, 0.6), ma = -0.2),
  n = 300)
## Error in arima.sim(list(order = c(2, 0, 1), ar = c(-0.7, 0.6), ma = -0.2), : 'ar' p
```

[1] "'ar' part of model is not stationary"

Consider the AR polynomial of

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2$$

The required condition for causality is that both the roots of the characteristic equation

$$\phi(z) = 1 - \phi_1 z - \phi_2 z^2 = 0$$

denoted by z_1 and z_2 are outside the unit circle.

Suppose that we aim to determine the equivalent required condition for causality in terms of ϕ_1 and ϕ_2 .

(d) Let us divide the characteristic equation by z^2 and obtain

$$\frac{1}{z^2} - \frac{\phi_1}{z} - \phi_2 = 0$$

If we define $y = \frac{1}{z}$, then the resulting equation can be expressed as

$$\alpha(y) = y^2 - \phi_1 y - \phi_2 = 0$$

Write down the equivalent required condition for causality in terms of the characteristic roots y_1 and y_2 and express ϕ_1 and ϕ_2 in terms of y_1 and y_2 . (3 marks)

$$|y_1| = \left| \frac{1}{z_1} \right| < 1, \quad |y_2| = \left| \frac{1}{z_2} \right| < 1$$

$$\begin{aligned} \alpha(y) &= (y - y_1)(y - y_2) = y^2 - (y_1 + y_2)y + y_1 y_2 = y^2 - \phi_1 y - \phi_2 = 0 \\ \Rightarrow \phi_1 &= y_1 + y_2, \quad \phi_2 = -y_1 y_2 \end{aligned}$$

(e) Following from part (d), determine the equivalent required condition for causality in terms of ϕ_1 and ϕ_2 . Hint: consider (i) the **sign** of $(y_1 + a)(y_2 + b)$, (ii) the **sign** of $(y_1 + c)(y_2 + d)$ and (iii) the magnitude of $y_1 y_2$ for suitably chosen constants a, b, c and d . (3 marks)

$$(y_1 - 1)(y_2 - 1) = y_1 y_2 - (y_1 + y_2) + 1 > 0$$

$$y_1 + y_2 - y_1 y_2 < 1$$

$$\Rightarrow \phi_1 + \phi_2 < 1$$

$$(y_1 + 1)(y_2 + 1) = y_1 y_2 + (y_1 + y_2) + 1 > 0$$

$$-(y_1 + y_2) - y_1 y_2 < 1$$

$$\Rightarrow -\phi_1 + \phi_2 < 1$$

$$|y_1 y_2| < 1$$

$$\Rightarrow |\phi_2| < 1$$

(f) Following from part (e), determine whether the model given above

$$x_t = -0.7x_{t-1} + 0.6x_{t-2} + w_t - 0.2w_{t-1}$$

is causal without finding the characteristic roots. (1 mark)

We have $\phi_1 = -0.7, \phi_2 = 0.6$ and so $|\phi_2| < 1, \phi_2 + \phi_1 = -0.1 < 1$ but $\phi_2 - \phi_1 = 1.3 > 1$, so the model is not causal.

Question 6

Consider the following two time series:

$$x_t = w_t + \alpha_1 w_{t-1} + \alpha_2 w_{t-2},$$

and

$$y_t = w_t + \beta_1 w_{t-1} + \beta_2 w_{t-2},$$

where w_t is white noise with variance σ_w^2 .

Furthermore, we have that

$$\begin{aligned}\alpha_1 &= 0.8465; \\ \alpha_2 &= 0.2476; \\ \beta_1 &= 0.8028; \\ \beta_2 &= 0.2498; \\ \sigma_w^2 &= 1.3647.\end{aligned}$$

Explain whether each of the following statements is true or false.

- (a) Both series are stationary and causal. (1 mark)
- (b) The variance of y_t is 2.3294. (2 marks)
- (c) The autocorrelation of x_t at lag 1 is 0.6152. (2 marks)
- (d) The cross-covariance $\gamma_{xy}(t, t+1) = 1.4265$. (2 marks)
- (a) True - because both are pure MA model with an AR polynomial of $\phi(B) = 1$.
- (b) True - because we have that $\gamma_y(t, t) = cov(y_t, y_t) = cov(w_t + \beta_1 w_{t-1} + \beta_2 w_{t-2}, w_t + \beta_1 w_{t-1} + \beta_2 w_{t-2}) = \sigma_w^2 (1 + \beta_1^2 + \beta_2^2)$
- (c) False - because we have $\gamma_x(t, t) = cov(x_t, x_t) = cov(w_t + \alpha_1 w_{t-1} + \alpha_2 w_{t-2}, w_t + \alpha_1 w_{t-1} + \alpha_2 w_{t-2}) = \sigma_w^2 (1 + \alpha_1^2 + \alpha_2^2)$ and $\gamma_x(t+1, t) = Cov(x_{t+1}, x_t) = Cov(w_{t+1} + \alpha_1 w_t + \alpha_2 w_{t-1}, w_t + \alpha_1 w_{t-1} + \alpha_2 w_{t-2}) = (\alpha_1 + \alpha_2 \alpha_1) \sigma_w^2 \Rightarrow \rho_x(1) = \frac{\gamma_x(t, t+1)}{\gamma_x(t, t)} = 0.594$
- (d) False - because we have that $\gamma_{xy}(t, t+1) = Cov(x_t, y_{t+1}) = Cov(w_t + \alpha_1 w_{t-1} + \alpha_2 w_{t-2}, w_{t+1} + \beta_1 w_t + \beta_2 w_{t-1}) = \sigma_w^2 (\beta_1 + \alpha_1 \beta_2) = 1.3841$

15 Appendix E - Practice MST Questions and Solutions

refer to next page

1. Aggregate claims from a risk have a compound Poisson distribution with Poisson parameter 50, and individual claim sizes X 's are distributed according to the following cumulative distribution function

$$F(x) = 1 - 0.4e^{-0.01x} - 0.6e^{-0.02x}, \quad x \geq 0.$$

- (a) Show that the individual claim sizes follow a mixed distribution in the form of

$$X = \alpha X_1 + \beta X_2$$

where both X_1 and X_2 are independent to each other. Determine the distribution of both X_1 and X_2 and their associated parameter(s) as well as parameters α and β . (3 marks)

$$f(x) = 0.004e^{-0.01x} + 0.012e^{-0.02x}$$

$$X = 0.4X_1 + 0.6X_2,$$

$$X_1 \sim \text{Exp}(0.01), \quad X_2 \sim \text{Exp}(0.02)$$

- (b) Calculate the value of $P(S \leq 4500)$ using a normal approximation. (4 marks)

$$m_1 = \frac{0.4}{0.01} + \frac{0.6}{0.02} = 70$$

$$m_2 = 2 \left(\frac{0.4}{0.01^2} + \frac{0.6}{0.02^2} \right) = 11000$$

$$m_3 = 6 \left(\frac{0.4}{0.01^3} + \frac{0.6}{0.02^3} \right) = 2850000$$

$$E(S) = 50(70) = 3500$$

$$\text{Var}(S) = 50(11000) = 550000$$

$$P(S \leq 4500) = P\left(Z \leq \frac{4500 - 3500}{\sqrt{550000}}\right)$$

$$= P(Z \leq 1.3484) = 0.911$$

- (c) Suppose that we use a compound negative binomial model to study the aggregate claims instead, where the mean of claims count $E(N)$ under the negative binomial distribution is the same as the mean of claims count under the Poisson distribution.

Without performing any numerical calculations, explain whether the corresponding approximate value of $P(S \leq 4500)$ using a normal approximation is larger than, equal to, or smaller than your answer in part (b). (3 marks)

Both models have identical $E(S) = E(N)E(X) = 3500$.

The negative binomial model features over-dispersion where $\text{Var}(N) > E(N)$, whereas the Poisson model has equi-dispersion $\text{Var}(N) = E(N)$. Hence, the negative binomial model has a larger $\text{Var}(S) = E(N)\text{Var}(X) + \text{Var}(N)(E(X))^2$ than that of the Poisson model.

Therefore we are effectively finding $P(Z < \text{smaller positive magnitude})$ and we obtain a smaller corresponding $P(S \leq 4500)$.

2. Consider the Archimedean generator

$$\psi_{\alpha,\beta}(t) = (t^{-\alpha} - 1)^\beta, \quad \text{for } \alpha > 0, \beta \geq 1.$$

satisfying the definition of Archimedean copula in which

$$C_{\alpha,\beta}(u, v) = \psi_{\alpha,\beta}^{-1}(\psi_{\alpha,\beta}(u) + \psi_{\alpha,\beta}(v)),$$

where we can easily see that $\psi_{\alpha,\beta}(1) = 0$, $\psi'_{\alpha,\beta}(t) < 0$ and $\psi''_{\alpha,\beta}(t) > 0$

- (a) Without determining the inverse of the generator function, derive an expression for the copula $C_{\alpha,\beta}(u, v)$ in terms of u and v . (2 marks)

From

$$\psi_{\alpha,\beta}(C_{\alpha,\beta}(u, v)) = \psi_{\alpha,\beta}(u) + \psi_{\alpha,\beta}(v),$$

we obtain

$$(C_{\alpha,\beta}(u, v)^{-\alpha} - 1)^\beta = (u^{-\alpha} - 1)^\beta + (v^{-\alpha} - 1)^\beta$$

and hence

$$C_{\alpha,\beta}(u, v) = \left\{ \left[(u^{-\alpha} - 1)^\beta + (v^{-\alpha} - 1)^\beta \right]^{1/\beta} + 1 \right\}^{-1/\alpha}.$$

- (b) Show that the copula

$$C(u, v) = \frac{uv}{u + v - uv}$$

belongs to the family of copulas $C_{\alpha,\beta}(u, v)$ defined by the generator $\psi_{\alpha,\beta}(t)$. Show your steps clearly. (3 marks)

Notice how the exponent powers of u and v in $C(u, v)$ are 1, so we try $\beta = 1$ and $\alpha = 1$ and obtain

$$\begin{aligned} C_{1,1}(u, v) &= \left\{ \left[(u^{-1} - 1)^1 + (v^{-1} - 1)^1 \right]^{1/1} + 1 \right\}^{-1/1} \\ &= \frac{1}{1/u + 1/v - 1} \\ &= \frac{uv}{v + u - uv}. \end{aligned}$$

This is the same copula as the implicit one derived in the lecture.

- (c) Assume that we can extend the parameter range of $C_{\alpha,\beta}(u, v)$ to include $\alpha = 0$. Identify $C_{0,1}(u, v)$, a well known copula. (3 marks)

$$\begin{aligned} C_{0,1}(u, v) &= \lim_{\alpha \rightarrow 0} (u^{-\alpha} + v^{-\alpha} - 1)^{-1/\alpha} \\ &= \lim_{\alpha \rightarrow 0} \left(\frac{(uv)^\alpha}{v^\alpha + u^\alpha - (uv)^\alpha} \right)^{1/\alpha} \\ &= \lim_{\alpha \rightarrow 0} \frac{uv}{(v^\alpha + u^\alpha - (uv)^\alpha)^{-1/\alpha}} \\ &= \frac{uv}{1} = uv, \end{aligned}$$

that is, we obtain the independence copula.

3. Suppose that the random variable X has the following probability density function

$$f_X(x) = \alpha e^{-\alpha x}, \quad x > 0$$

for parameter $\alpha > 0$.

- (a) Determine the probability density function of $Y = e^X$. (2 marks)

$$\begin{aligned} P(X \leq x) &= \int_0^x \alpha e^{-\alpha z} dz = 1 - e^{-\alpha x} \\ P(Y \leq y) &= P(e^X \leq y) \\ &= P(X \leq \log y) = 1 - y^{-\alpha}, \quad y > 1 \\ f_Y(y) &= \alpha y^{-\alpha-1}, \quad y > 1 \end{aligned}$$

- (b) By considering the hazard rate function of both X and Y , compare their tail weight. (3 marks)

$$\begin{aligned} h(x) &= \frac{f(x)}{S(x)} \\ &= \frac{\alpha e^{-\alpha x}}{1 - (1 - e^{-\alpha x})} = \alpha \\ h(y) &= \frac{f(y)}{S(y)} \\ &= \frac{\alpha y^{-\alpha-1}}{y^{-\alpha}} = \frac{\alpha}{y} \end{aligned}$$

Since $h(x)$ is constant and $h(y)$ is a decreasing function of y , Y has a heavier tail weight than X .

- (c) Keeping the value of α fixed, how does the survival function $S(y) = P(Y > y)$ behave as $y \rightarrow \infty$? Discuss. (1 mark)

We have $S(y) = 1 - (1 - y^{-\alpha}) = y^{-\alpha}$, so the survival function decays to zero according to the magnitude of α .

- (d) Determine the derivative of $S(y)$ with respect to α . What can you conclude from the result? (2 marks)

$$\begin{aligned} \frac{dS(y)}{d\alpha} &= y^{-\alpha} \log \frac{1}{y} \\ &= -\log y \times y^{-\alpha} < 0 \end{aligned}$$

So the larger the value of α , the smaller the value of $S(y)$, that is, the survival function decays faster (i.e., having a lighter tail, other things being equal).

- (e) Define random variable $Z = Y - 1$. Show that Z belongs to the Generalised Pareto distribution (GPD) and identify the associated parameters. (2 marks)

$$\begin{aligned} P(Z \leq z) &= P(Y \leq z + 1) \\ &= 1 - (z + 1)^{-\alpha} = 1 - \left(1 + \gamma \frac{z}{\sigma}\right)^{-\frac{1}{\gamma}} = G_{\gamma, \sigma}(z) \end{aligned}$$

where we have $\sigma = \gamma = \frac{1}{\alpha}$.

- (f) By considering the survival function $S(y)$, comment on whether Y is of Generalised Extreme Value (GEV) distribution of Fréchet type by examining the definition of tail index in

$$S(y) = y^{-\frac{1}{\gamma}} L(y)$$

for slowly varying function $L(y)$. (1 mark)

Since we can write $S(y) = y^{-\alpha}$, it follows immediately that the tail index is given by $-\frac{1}{\gamma} = -\alpha$, that is, Y is of Fréchet type with shape parameter $\gamma = \frac{1}{\alpha} > 0$.

END OF TEST

16 Appendix F - MST Questions and Solutions

refer to next page

1. Suppose that the individual ground up loss of an insured, denoted by X , follows a uniform distribution between 0 and 1200. The insurance policy has a deductible level of $d > 0$, such that if the loss amount is below d , the insured will not be able to claim from the insurer. If the loss amount is above d , the insurer will pay the excess amount over d denoted by $X - d$, subjected to a maximum claim payment of $M < 1200 - d$.

- (a) Derive the mean and the variance of the insurance payout under the policy in terms of d and M . (5 marks)

$$Y = (X \wedge (d + M)) - (X \wedge d) = \begin{cases} 0, & 0 < X \leq d \\ X - d, & d < X < d + M \\ M, & d + M \leq X < 1200 \end{cases}$$

$$f_X(x) = \frac{1}{1200}$$

$$\begin{aligned} E(Y) &= \frac{1}{1200} \int_d^{d+M} (x - d) dx + \frac{1}{1200} \int_{d+M}^{1200} M dx \\ &= \frac{1}{1200} \frac{M^2}{2} + \frac{M}{1200} (1200 - d - M) \\ &= M - \frac{1}{1200} \left(M d + \frac{M^2}{2} \right) \end{aligned}$$

$$\begin{aligned} \text{OR } &= \int_0^{d+M} x f_X(x) dx + (d + M) (1 - F_X(d + M)) - \left(\int_0^d x f_X(x) dx + d (1 - F_X(d)) \right) \\ &= \left(\frac{(d + M)^2}{2(1200)} + (d + M) \times \frac{1200 - (d + M)}{1200} \right) - \left(\frac{d^2}{2(1200)} + d \times \frac{1200 - d}{1200} \right) \\ &= (d + M) - \frac{(d + M)^2}{2(1200)} - \left(d - \frac{d^2}{2(1200)} \right) \\ &= M - \frac{1}{1200} \left(M d + \frac{M^2}{2} \right) \end{aligned}$$

$$\begin{aligned} \text{Var}(Y) &= E(Y^2) - (E(Y))^2 \\ &= \frac{1}{1200} \int_d^{d+M} (x - d)^2 dx + \frac{1}{1200} \int_{d+M}^{1200} M^2 dx - \left(M - \frac{1}{1200} \left(M d + \frac{M^2}{2} \right) \right)^2 \\ &= \frac{1}{1200} \left(\frac{M^3}{3} + M^2 (1200 - d - M) \right) - \left(M - \frac{1}{1200} \left(M d + \frac{M^2}{2} \right) \right)^2 \end{aligned}$$

- (b) Is the maximum random loss to be borne by the insured after taking into

account the random claim payment by the insurer equal to the deductible level d ? Explain your answer clearly. (1 mark)

No, the random loss is given by $X \wedge d + (X - M - d)_+$, which equals to $X - M > d$ for large losses $d + M < X < 1200$.

- (c) Suppose that the insurer sold 500 such policies, each with $d = 60$ and $M = 800$.

The claim frequency under each policy follows a Poisson distribution with a mean of 0.8. Using the Normal approximation, calculate the probability that the total aggregate claims from all 500 policies is greater than 125% of the pure premium received from the policies. (4 marks)

$$\begin{aligned} E(Y) &= 800 - \frac{1}{1200} \left(800(60) + \frac{800^2}{2} \right) \\ &= \frac{1480}{3} = 493.333 \end{aligned}$$

$$\begin{aligned} \text{Var}(Y) &= \frac{1}{1200} \left(\frac{800^3}{3} + 800^2(1200 - 60 - 800) \right) - \left(\frac{1480}{3} \right)^2 \\ &= \frac{721600}{9} = 80177.778 \end{aligned}$$

$$\begin{aligned} E(S) &= E(N)E(Y) = 0.8(500)E(Y) \\ &= \frac{592000}{3} = 197333.333 \end{aligned}$$

$$\begin{aligned} \text{Var}(S) &= E(N)\text{Var}(Y) + (E(Y))^2\text{Var}(N) = 400\text{Var}(Y) + 400(E(Y))^2 \\ \text{OR } m_2 &= 400E(Y^2) = 400 \times \frac{1}{1200} \left(\frac{800^3}{3} + 800^2(1200 - 60 - 800) \right) \\ &= \frac{1164800000}{9} = 129422222.222 \end{aligned}$$

$$\begin{aligned} \Pr(S > 1.25E(S)) &= \Pr \left(Z > \frac{1.25E(S) - E(S)}{\sqrt{\text{Var}(S)}} \right) \\ &= \Pr(Z > 4.336467) = \Pr(Z > 4.34) \\ &\approx 0 \end{aligned}$$

Overall, the performance in this question was poorer than expected.

- Part (a) was poorly answered. Common mistakes include not accounting for the upper limit M , incorrectly using M as the upper bound for $Y = X - d$, incorrect

subsequent steps in deriving the mean, incorrect (part of) initial integral such as using $\int_d^{d+M} \frac{x}{1200} dx$, did not realise that $E(((X \wedge (d+M)) - (X \wedge d))^2) \neq E((X \wedge (d+M))^2) + E((X \wedge d)^2)$, probably due to time constraint / pressure.

- The performance in part (b) was mixed. Common mistakes include not explaining how / why the loss is more than d , incorrectly using the deductible d as the reason, or simply did not provide any explanation.
- Part (c) was well answered, considering the performance in part (a), which **emphasizes the importance of not ignoring subsequent parts even if you know that you did not manage to solve the earlier parts correctly**. The formula for the mean and the variance of the aggregate sum is straightforward and the same goes to the probability calculation. Common mistakes include not using 500 policies into the calculation, mis-interpretation of pure premium and not calculating the numerical values etc.

2. You are given the following joint distribution function for bivariate random variables X_1 and X_2

$$\Pr(X_1 \leq x_1, X_2 \leq x_2) = 1 - (1 - (1 - e^{-3x_1\theta}) (1 - (1 - x_2)^\theta))^{\frac{1}{\theta}}, \quad x_1 > 0, 0 < x_2 < 1,$$

for $\theta \geq 1$.

- Determine and identify the underlying marginal distributions of X_1 and X_2 respectively and their associated parameters. Note that simply writing down the marginal cumulative distribution function is not sufficient. (3 marks)

$$F_{X_1}(x_1) = \Pr(X_1 \leq x_1, X_2 \leq 1) = 1 - e^{-3x_1}$$

$$\Rightarrow X_1 \sim \text{Exp}(3), \quad X_1 > 0$$

$$F_{X_2}(x_2) = \Pr(X_1 \leq \infty, X_2 \leq x_2) = x_2$$

$$\Rightarrow X_2 \sim U(0, 1), \quad 0 < X_2 < 1$$

(b) Determine the copula function $C(u_1, u_2)$ in terms of u_1 and u_2 . (3 marks)

$$\begin{aligned} F_{X_1}(x_1) &= 1 - e^{-3x_1} = u_1 \\ \Rightarrow x_1 &= -\frac{\log(1 - u_1)}{3} \\ F_{X_2}(x_2) &= x_2 = u_2 \\ C(u_1, u_2) &= 1 - \left(1 - \left(1 - e^{-3\theta \times -\frac{\log(1 - u_1)}{3}}\right) \left(1 - (1 - u_2)^\theta\right)\right)^{\frac{1}{\theta}} \\ &= 1 - \left(1 - (1 - (1 - u_1)^\theta)\right) \left(1 - (1 - u_2)^\theta\right)^{\frac{1}{\theta}} \end{aligned}$$

(c) The copula function in part (b) can be used to link any two marginal distributions (i.e., not necessarily of the same form as those obtained in part (a)). True or false? Explain. (1 mark)

True. Copula is flexible in that it can take in any marginal cumulative distribution functions and express the joint cumulative distribution function via its explicit form in terms of both marginals u_1 and u_2 .

(d) What is the property of the copula if $\theta = 1$? (2 marks)

We obtain $C(u_1, u_2) = u_1 u_2$, which is an independence copula.

(e) Following from the expression in part (b), explain **in words** whether the copula is an Archimedean copula. You do not need to work out the mathematical formulation or details. Instead, describe your thought process and be as specific as possible. (2 marks)

The product of $(1 - (1 - u_1)^\theta)$ and $(1 - (1 - u_2)^\theta)$ can be expressed in terms of the addition of two generator functions so long as we take logarithm before we perform the addition and we invert the operation by taking $\exp(\cdot)$ through its inverse function (i.e., the inverse of the generator function) afterwards.

Overall, the performance in this question was in line with expectation.

- Part (a) was well answered in general. Common mistakes include not simplifying the expression further (to cancel out the impact of θ), stating incorrect parameters for the distributions and in some cases identify the wrong distributions.

- The performance in part (b) was a little below expectation. Common mistakes include careless mistakes due to the use of many brackets in the original joint cdf that result in final incorrect expression.
- Part (c) was poorly answered given its simplicity. The keyword here is the **copula function in part (b) is in terms of u and v** and not in terms of x_1 and x_2 . Common mistakes include incorrectly arguing the function is only valid for the given distribution in the question, the range of values for x_1 and x_2 etc, the existence of unique copula for each joint cdf etc.
- Part (d) was well answered and errors carried forward receive full credit in many cases. However, many students state the answer as independence copula even when their correct simplification on an incorrect earlier part did not suggest so. Such attempt should be avoided in general, especially in the workplace. It is perfectly fine to follow the process and get an incorrect result based on earlier errors, than say arguing for the correct final outcome, because “two wrongs don’t make a right”.
- Part (e) was in line with expectation. This is a tricky question. Common mistakes include not realising we can use generator to turn the product into summation that satisfies the definition of Archimedean copula. Many students simply stated the conditions without analysing the copula function.

3. Suppose that the random variable X is uniformly distributed between 0 and 1.

(a) Derive the cumulative distribution function of random variable Y defined as

$$Y = -2025 \log X.$$

(2 marks)

$$F_X(x) = x$$

$$\begin{aligned} \Pr(Y \leq y) &= \Pr(-2025 \log X \leq y) = \Pr(X \geq e^{-\frac{y}{2025}}) \\ &= 1 - e^{-\frac{y}{2025}}, \quad y > 0 \end{aligned}$$

- (b) Following from part (a), we denote the complement of the threshold exceedance probability as

$$\bar{F}_u(x) = 1 - F_u(x) = \Pr(Y - u > x | Y > u)$$

Show that $\bar{F}_u(x) = 1 - G_{\gamma,\sigma}(x)$ for suitably chosen parameters γ and σ of a Generalised Pareto distribution (GPD) G . (3 marks)

$$\begin{aligned}\bar{F}_u(x) &= 1 - \frac{F(x+u) - F(u)}{1 - F(u)} = \frac{1 - F(x+u)}{1 - F(u)} \\ &= \frac{1 - \left(1 - e^{-\frac{x+u}{2025}}\right)}{1 - \left(1 - e^{-\frac{u}{2025}}\right)} \\ &= e^{-\frac{x}{2025}} = 1 - \left(1 - e^{-\frac{x}{2025}}\right) = 1 - G_{\gamma,\sigma}(x)\end{aligned}$$

for $\gamma = 0$ and $\sigma = 2025$.

- (c) Consider independently and identically distributed (iid) random variables $Y_i, i = 1, 2, \dots, n$ with distribution function obtained in part (a). Show that the limiting distribution of the normalized n -block maximum is from the Generalised Extreme Value (GEV) distribution. Show your steps clearly and specify the range of parameter values for γ . (4 marks)

$$\begin{aligned}\lim_{n \rightarrow \infty} \Pr\left(\frac{Y_{1,n} - b_n}{a_n} \leq y\right) &= \lim_{n \rightarrow \infty} \Pr(Y_{1,n} \leq a_n y + b_n) \\ &= \lim_{n \rightarrow \infty} F^n(a_n y + b_n) \\ &= \lim_{n \rightarrow \infty} \left(1 - e^{-\left(\frac{a_n y + b_n}{2025}\right)}\right)^n \\ &= \lim_{n \rightarrow \infty} \left(1 - \frac{e^{-y}}{n}\right)^n = e^{-e^{-y}}\end{aligned}$$

by choosing normalizing constants $a_n = 2025, b_n = 2025 \log n, \mu = 0, \sigma = 1$, where $x \in \mathbb{R}$. We have $\gamma = 0$, i.e., the GEV of the Gumbel type.

Overall, the performance in this question was better than expected.

- Part (a) was well answered. Common mistakes include forgetting to switch the sign of the inequality when they divided both sides by -2025 .

- Part (b) was very well answered. A number of students incorrectly used the cdf of uniform distribution instead of their cdf obtained in part (a).
- Part (c) was well answered in general. Common mistakes include not able to determine a_n and b_n due to some mistakes in algebra or struggle to understand the question. Partial marks were awarded to students who did not show full working but able to provide relevant information to GEV and correctly stated $\gamma = 0$.

17 Appendix G - Assignment R Code and Output

refer to next page

ACTL30007 - Omar Amin - Assignment 1

Omar

2025-06-14

R Markdown

```
library(readxl)
library(tidyr)
library(dplyr)

## 
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
## 
##     filter, lag

## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union

# Load data
df1 = read_excel("1463797.xlsx", sheet=1)
df2 = read_excel("1463797.xlsx", sheet=2)
df3 = read_excel("1463797.xlsx", sheet=3)

# Filter strictly positive claims
sev1 <- df1$claim_size[df1$claim_size > 0]
sev2 <- df2$claim_size[df2$claim_size > 0]
sev3 <- df3$claim_size[df3$claim_size > 0]

data <- list(df1, df2, df3)
```

Including Plots

You can also embed plots, for example:

```
## Loading required package: MASS

## 
## Attaching package: 'MASS'
```

```

## The following object is masked from 'package:dplyr':
##
##     select

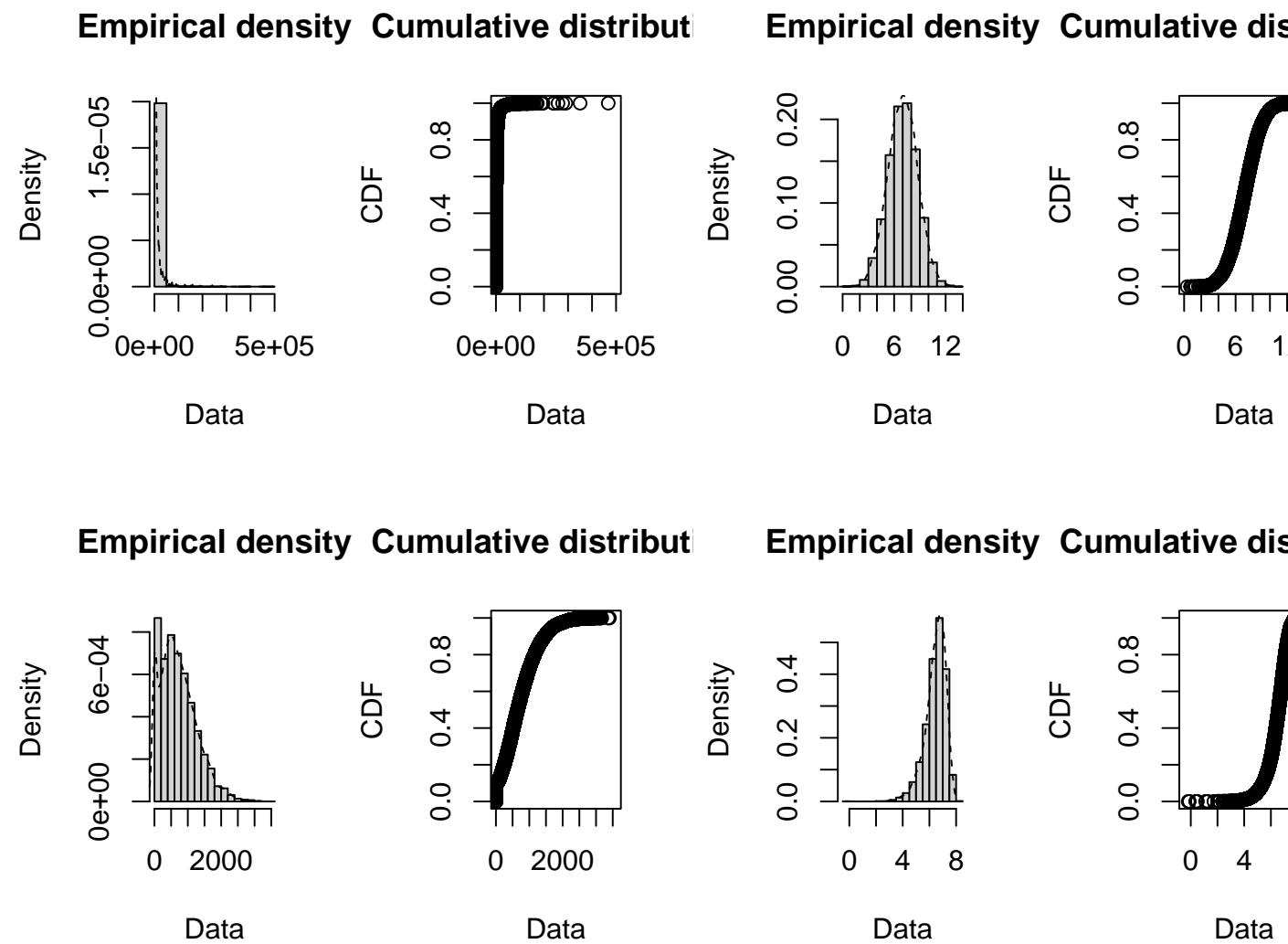
## Loading required package: survival

## Portfolio 1: 68.641%

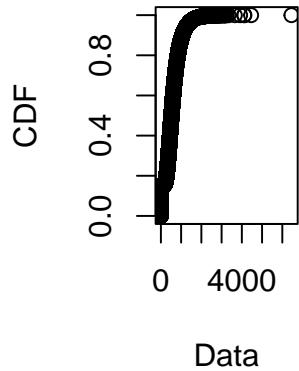
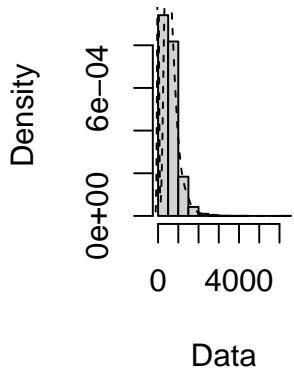
## Portfolio 2: 90.432%

## Portfolio 3: 84.787%

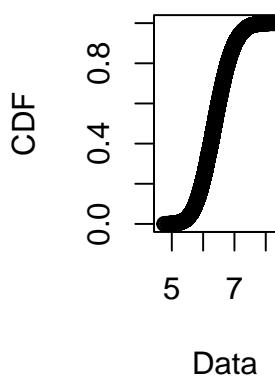
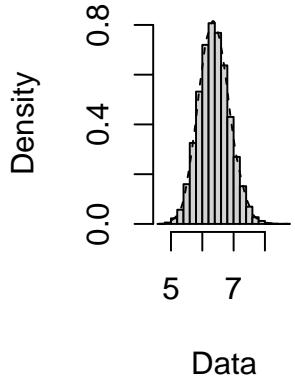
```



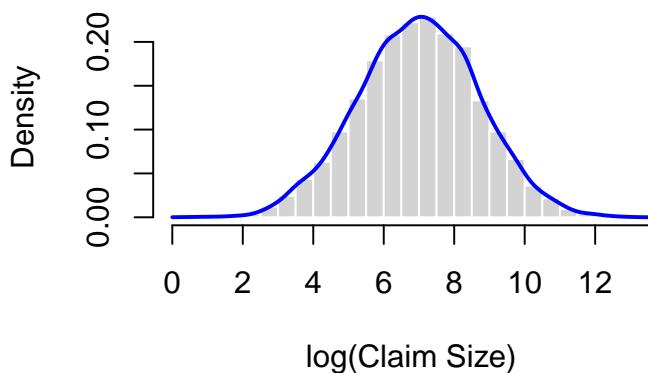
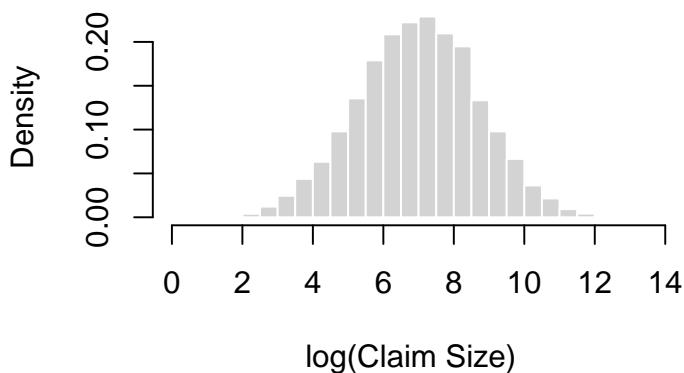
Empirical density Cumulative distribution function



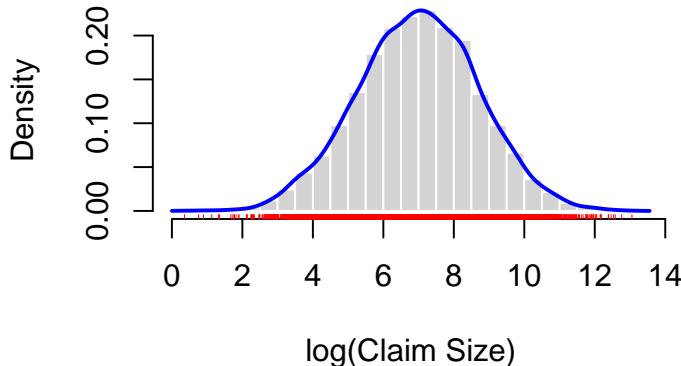
Empirical density Cumulative distribution function



Portfolio 1: Log–Claim Sizes



Portfolio 1: Log–Claim Sizes



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
summarize_portfolio <- function(df) {  
  count_data <- df %>%  
    group_by(id) %>%  
    summarise(n_claims = ifelse(length(claim_size) == 1 & all(claim_size == 0), 0L, n()))  
  
  zero_prop <- mean(count_data$n_claims == 0)  
  list(  
    zero_prop = zero_prop,  
    mean = mean(count_data$n_claims),  
    var = var(count_data$n_claims),  
    data = count_data  
)  
}  
  
stats1 <- summarize_portfolio(df1)  
stats2 <- summarize_portfolio(df2)  
stats3 <- summarize_portfolio(df3)  
stats1; stats2; stats3  
  
## $zero_prop  
## [1] 0.413  
##  
## $mean  
## [1] 0.904  
##  
## $var  
## [1] 0.9426783  
##  
## $data  
## # A tibble: 10,000 x 2  
##       id n_claims  
##   <dbl>     <int>
```

```

## 1      1      0
## 2      2      2
## 3      3      2
## 4      4      0
## 5      5      0
## 6      6      1
## 7      7      0
## 8      8      1
## 9      9      0
## 10     10     0
## # i 9,990 more rows

## $zero_prop
## [1] 0.1692
##
## $mean
## [1] 1.5992
##
## $var
## [1] 1.299889
##
## $data
## # A tibble: 10,000 x 2
##       id n_claims
##   <dbl>    <int>
## 1     1         2
## 2     2         2
## 3     3         2
## 4     4         1
## 5     5         3
## 6     6         3
## 7     7         2
## 8     8         1
## 9     9         2
## 10    10        2
## # i 9,990 more rows

## $zero_prop
## [1] 0.2923
##
## $mean
## [1] 1.6291
##
## $var
## [1] 2.179151
##
## $data
## # A tibble: 10,000 x 2
##       id n_claims
##   <dbl>    <int>
## 1     1         2
## 2     2         0
## 3     3         4
## 4     4         3

```

```

## 5      5      6
## 6      6      0
## 7      7      0
## 8      8      1
## 9      9      0
## 10     10     0
## # i 9,990 more rows

cat("Index of Dispersions P1, P2, P3: ", c(stats1$var/stats1$mean, stats2$var/stats2$mean, stats3$var/
stats3$mean))

## Index of Dispersions P1, P2, P3: 1.042786 0.8128373 1.337641

# Fit Poisson and (if appropriate) Negative Binomial for claim counts
library(MASS)
df1_counts <- df1 %>% group_by(id) %>% summarise(count = if_else(n() == 1 & any(claim_size == 0), 0L, as.in-
df2_counts <- df2 %>% group_by(id) %>% summarise(count = if_else(n() == 1 & any(claim_size == 0), 0L, as.in-
df3_counts <- df3 %>% group_by(id) %>% summarise(count = if_else(n() == 1 & any(claim_size == 0), 0L, as.in-

# Frequency models (intercept-only, since no covariates)
model1_pois <- glm(count ~ 1, data = df1_counts, family = "poisson")
model2_pois <- glm(count ~ 1, data = df2_counts, family = "poisson")
model3_pois <- glm(count ~ 1, data = df3_counts, family = "poisson")
# Negative Binomial fit for portfolios showing overdispersion
model3_nb <- glm.nb(count ~ 1, data = df3_counts)           # (only portfolio3 needs NB significantly)
coef(model3_nb); model3_nb$theta                            # view NB parameters (intercept log-mean, the

## (Intercept)
## 0.4880277

## [1] 3.862073

library(fitdistrplus)
library(actuar)  # for Pareto distribution functions

##
## Attaching package: 'actuar'

## The following objects are masked from 'package:stats':
## 
##     sd, var

## The following object is masked from 'package:grDevices':
## 
##     cm

sev1 <- df1$claim_size[df1$claim_size > 0]    # Portfolio1 claim sizes > 0
sev2 <- df2$claim_size[df2$claim_size > 0]
sev3 <- df3$claim_size[df3$claim_size > 0]

# Fit parametric severity distributions for Portfolio1 (heavy-tailed)

```

```

fit1_gamma <- fitdist(sev1, "gamma", start=list(shape=1, scale=mean(sev1)))    # Gamma
fit1_lnorm <- fitdist(sev1, "lnorm")                                         # Lognormal
fit1_pareto<- fitdist(sev1, "pareto", start=list(shape=2, scale=500))        # Pareto

gofstat(list(fit1_gamma, fit1_lnorm, fit1_pareto), fitnames = c("Gamma","Lognormal","Pareto"))$aic

##      Gamma Lognormal      Pareto
## 164796.8 161740.5 161931.7

fit1_gamma$estimate; fit1_lnorm$estimate; fit1_pareto$estimate                      # view parameter estimates

##      shape      scale
## 0.4543333 9873.8802195

##  meanlog      sdlog
## 6.989846 1.710560

##      shape      scale
## 1.096714 1257.609778

# ----- Frequency models for Portfolios 1 and 2 -----
# Prepare claim counts per policy
counts_df <- function(df) {
  df %>%
    group_by(id) %>%
    summarise(n_claims = ifelse(length(claim_size) == 1 & all(claim_size == 0), 0L, n()))
}

counts1 <- counts_df(df1)
counts2 <- counts_df(df2)

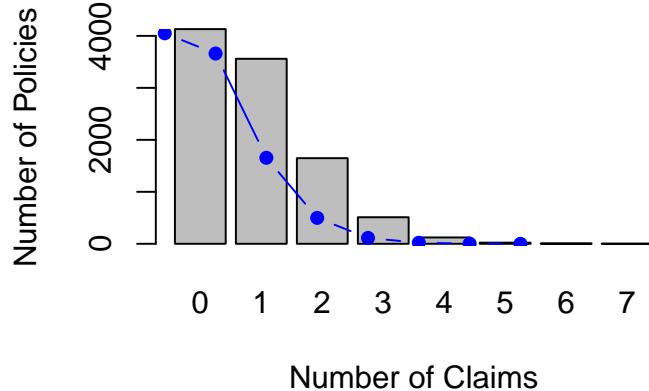
# Portfolio 1
fit_pois1 <- glm(n_claims ~ 1, family = poisson(), data = counts1)
lambda1 <- exp(coef(fit_pois1))
expected1 <- dpois(0:max(counts1$n_claims), lambda1) * nrow(counts1)
obs1 <- table(counts1$n_claims)

# Portfolio 2
fit_pois2 <- glm(n_claims ~ 1, family = poisson(), data = counts2)
lambda2 <- exp(coef(fit_pois2))
expected2 <- dpois(0:max(counts2$n_claims), lambda2) * nrow(counts2)
obs2 <- table(counts2$n_claims)

# Plot for Portfolio 1
barplot(obs1, col = "gray", names.arg = names(obs1),
         main = "Portfolio 1 Claim Count Fit (Poisson)",
         xlab = "Number of Claims", ylab = "Number of Policies")
lines(0:max(counts1$n_claims), expected1, col = "blue", type = "b", pch = 16)

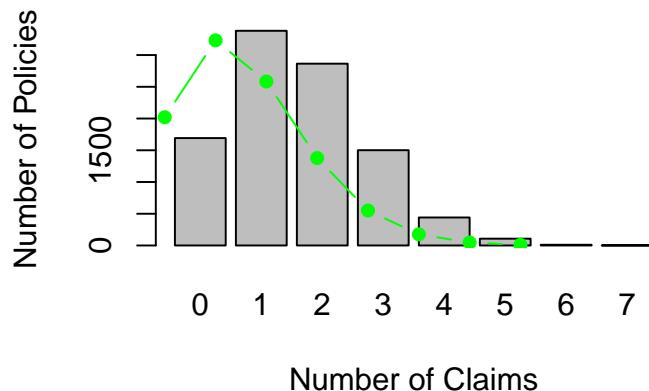
```

Portfolio 1 Claim Count Fit (Poisson)



```
# Plot for Portfolio 2
barplot(obs2, col = "gray", names.arg = names(obs2),
        main = "Portfolio 2 Claim Count Fit (Poisson)",
        xlab = "Number of Claims", ylab = "Number of Policies")
lines(0:max(counts2$n_claims), expected2, col = "green", type = "b", pch = 16)
```

Portfolio 2 Claim Count Fit (Poisson)



```
# ----- Frequency models for Portfolio 3 -----
# Fit Poisson and Negative Binomial
fit_pois3 <- glm(n_claims ~ 1, family = poisson(), data = stats3$data)
fit_nb3   <- glm.nb(n_claims ~ 1, data = stats3$data)

# Visualize model fit
obs_counts <- table(stats3$data$n_claims)
lambda3 <- exp(coef(fit_pois3))
```

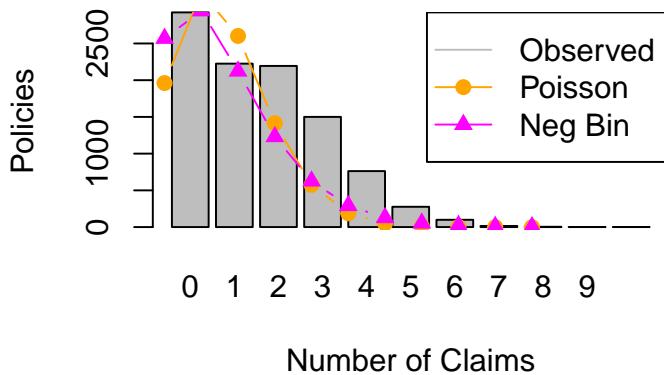
```

mu3 <- lambda3
theta3 <- fit_nb3$theta
expected_pois <- dpois(0:max(stats3$data$n_claims), lambda3) * nrow(stats3$data)
expected_nb   <- dnbinom(0:max(stats3$data$n_claims), mu = mu3, size = theta3) * nrow(stats3$data)

barplot(obs_counts, col = "gray", names.arg = names(obs_counts),
        main = "Portfolio 3 Claim Count", xlab = "Number of Claims", ylab = "Policies")
lines(0:max(stats3$data$n_claims), expected_pois, type = "b", col = "orange", pch = 19)
lines(0:max(stats3$data$n_claims), expected_nb,   type = "b", col = "magenta", pch = 17)
legend("topright", legend = c("Observed", "Poisson", "Neg Bin"),
       col = c("gray", "orange", "magenta"), pch = c(NA, 19, 17), lty = 1)

```

Portfolio 3 Claim Count



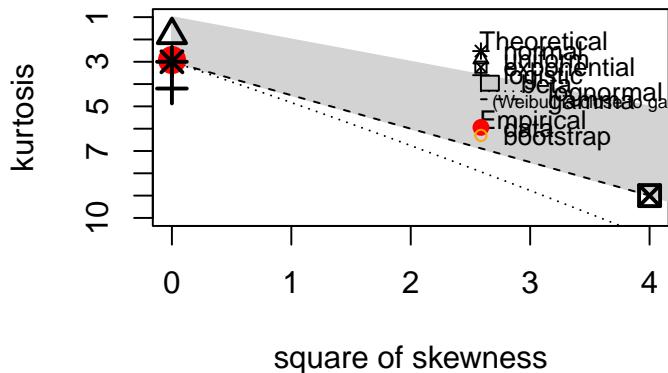
```

library(fitdistrplus)

# Portfolio 1 - log-transformed claim sizes
sev1 <- df1$claim_size[df1$claim_size > 0]
descdist(log(sev1), boot = 1000)

```

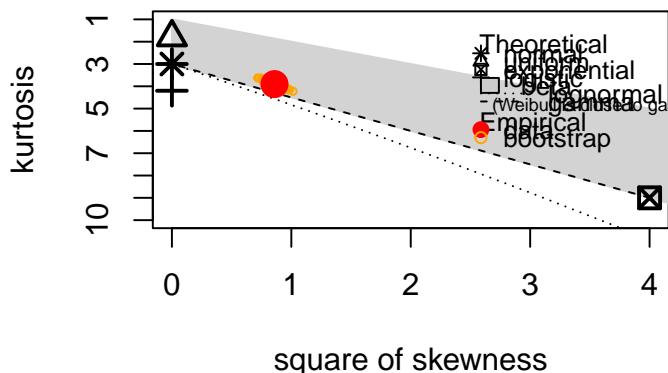
Cullen and Frey graph



```
## summary statistics
## -----
## min: 0.3577616 max: 13.05511
## median: 7.012698
## mean: 6.989846
## estimated sd: 1.710654
## estimated skewness: -0.02848522
## estimated kurtosis: 2.923099

sev2 <- df2$claim_size[df2$claim_size > 0]
descdist(sev2, boot = 1000)
```

Cullen and Frey graph



```
## summary statistics
## -----
## min: 0.8236877 max: 3401.825
```

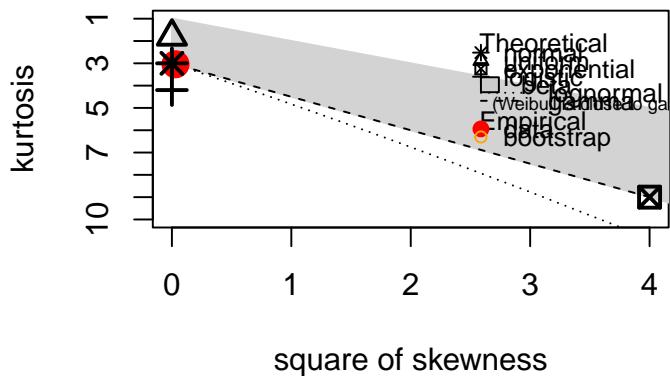
```

## median: 717.4712
## mean: 802.7218
## estimated sd: 506.1836
## estimated skewness: 0.9267448
## estimated kurtosis: 3.901759

sev3 <- df3$claim_size[df3$claim_size > 0]
descdist(log(sev3), boot = 1000)

```

Cullen and Frey graph



```

## summary statistics
## -----
## min: 4.75679   max: 8.772254
## median: 6.368575
## mean: 6.382347
## estimated sd: 0.4890409
## estimated skewness: 0.173933
## estimated kurtosis: 3.042674

#--- 1. Prepare log-severity vectors
logs1 <- log(df1$claim_size[df1$claim_size > 0])
logs2 <- log(df2$claim_size[df2$claim_size > 0])
logs3 <- log(df3$claim_size[df3$claim_size > 0])

#--- 2. A plotting function
plot_log_severity <- function(logs, title) {
  x_min <- floor(min(logs))
  x_max <- ceiling(max(logs)) + 0.5

  hist(logs,
        breaks = seq(x_min, x_max, by = 0.5),
        freq   = FALSE,
        xlim   = c(x_min, x_max),
        col    = "lightgray",
        border = "white",

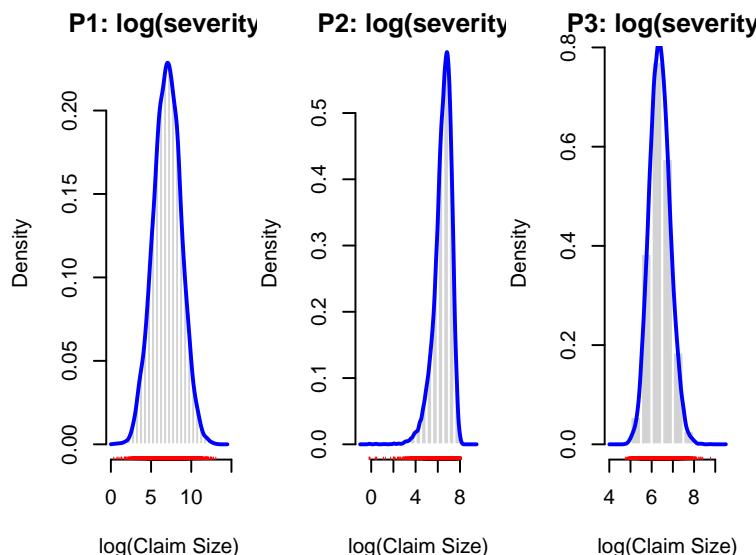
```

```

    main   = title,
    xlab   = "log(Claim Size)"
)
dens <- density(logs, from = x_min, to = x_max)
lines(dens, col="blue", lwd=2)
rug(logs, col="red", ticksize=0.02)
}

#--- 3. Plot side by side
par(mfrow = c(1,3), mar = c(4,4,2,1))
plot_log_severity(logs1, "P1: log(severity)")
plot_log_severity(logs2, "P2: log(severity)")
plot_log_severity(logs3, "P3: log(severity)")

```



```
par(mfrow = c(1,1))
```

Question 1: Summarize the recommended modelling parameters for each portfolio of risk.

```

# Load libraries & data
library(readxl); library(dplyr); library(fitdistrplus); library(MASS)

df_list <- lapply(1:3, function(i) read_excel("1463797.xlsx", sheet = i))

# Summarize frequency & choose model
freq_summary <- lapply(df_list, function(df) {
  counts <- df %>% group_by(id) %>% summarise(
    zero      = all(claim_size==0),
    claims    = ifelse(zero,0L,n())
  )
  m <- mean(counts$claims); v <- var(counts$claims)
  zero_pct <- 100*mean(counts$claims==0)
})

```

```

if (v/m > 1.05) {
  size <- m^2/(v-m); model <- "Negative Binomial"; params <- c(mu=m, size=size)
} else {
  model <- "Poisson"; params <- c(lambda=m)
}
list(zero_pct=round(zero_pct,1), model=model, params=round(params,3))
})

# Summarize severity & choose model
sev_summary <- lapply(df_list, function(df) {
  sev <- df$claim_size[df$claim_size > 0]

  # fit both
  gfit <- fitdist(sev, "gamma", start=list(shape=1, scale=mean(sev)))
  lfit <- fitdist(sev, "lnorm")

  # pull out AIC
  aic_g <- gofstat(gfit)$aic
  aic_l <- gofstat(lfit)$aic

  if (aic_l < aic_g) {
    model <- "Lognormal"
    params <- lfit$estimate[c("meanlog","sdlog")]
  } else {
    model <- "Gamma"
    params <- gfit$estimate[c("shape","scale")]
  }
  list(model = model, params = round(params,3))
})

# Build and print summary table
summary_tbl <- tibble::tibble(
  Portfolio      = 1:3,
  ZeroClaimPct = sapply(freq_summary, `[[`, "zero_pct"),
  FreqModel     = sapply(freq_summary, `[[`, "model"),
  FreqParams    = sapply(freq_summary, function(x) paste(names(x$params), "=" ,x$params, collapse="")),
  SevModel      = sapply(sev_summary, `[[`, "model"),
  SevParams     = sapply(sev_summary, function(x) paste(names(x$params), "=" ,x$params, collapse=""))
)
print(summary_tbl)

## # A tibble: 3 x 6
##   Portfolio ZeroClaimPct FreqModel          FreqParams      SevModel SevParams
##       <int>        <dbl> <chr>            <chr>           <chr>   <chr>
## 1         1        41.3 Poisson          lambda = 0.904 Lognorm~ meanlog ~
## 2         2        16.9 Poisson          lambda = 1.599  Gamma    shape = ~
## 3         3        29.2 Negative Binomial mu = 1.629, size ~ Lognorm~ meanlog ~

# Barplot of zero-claim %
library(ggplot2)

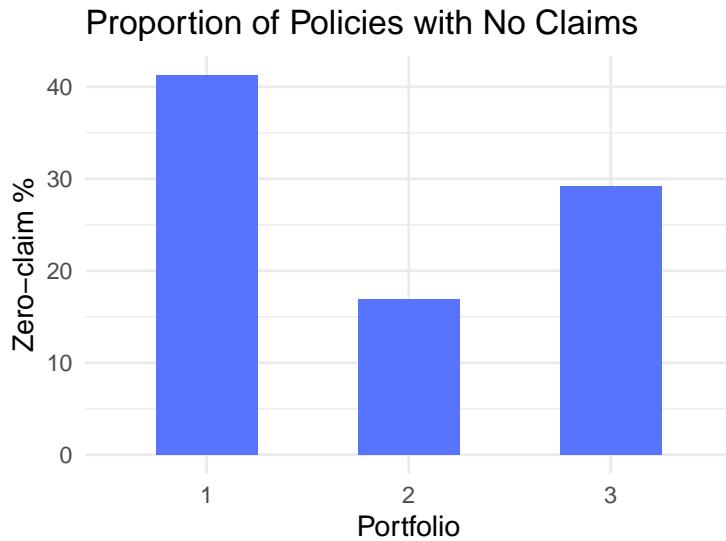
ggplot(summary_tbl, aes(x = factor(Portfolio), y = ZeroClaimPct)) +

```

```

geom_col(width = 0.5, fill = "#5874fc") +
labs(
  x = "Portfolio",
  y = "Zero-claim %",
  title = "Proportion of Policies with No Claims"
) +
theme_minimal()

```



Question 2

Part 1: Dispersion of the policies

```

library(dplyr)
library(ggplot2)

get_disp <- function(df, name) {
  counts <- df %>% group_by(id) %>%
    summarise(claims = ifelse(all(claim_size==0), 0L, n())) %>% ungroup()
  m <- mean(counts$claims); v <- var(counts$claims)
  data.frame(Portfolio = name, Dispersion = v/m)
}

disp1 <- get_disp(df1, "P1")
disp2 <- get_disp(df2, "P2")
disp3 <- get_disp(df3, "P3")
disp_df <- bind_rows(disp1, disp2, disp3)

ggplot(disp_df, aes(x = Portfolio, y = Dispersion, fill = Portfolio)) +
  geom_col(width = 0.3, show.legend = FALSE) +
  scale_x_discrete(
    expand = c(.3,0)
) +

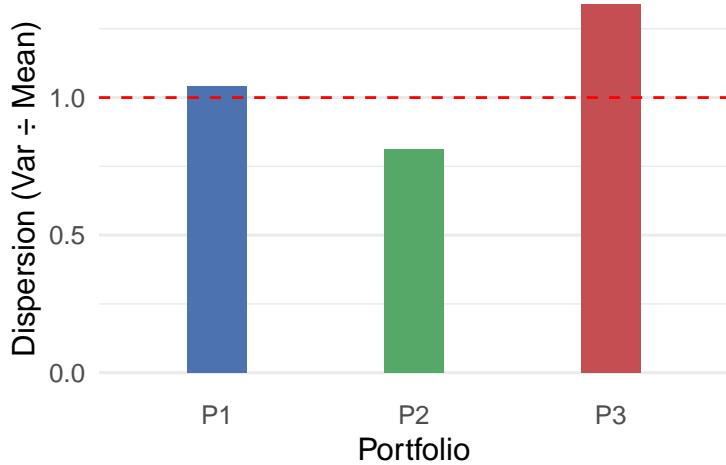
```

```

scale_fill_manual(values = c("P1" = "#4C72B0",
                           "P2" = "#55A868",
                           "P3" = "#C44E52")) +
geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
labs(
  x = "Portfolio",
  y = "Dispersion (Var ÷ Mean)",
  title = "Claim Count Dispersion by Portfolio"
) +
theme_minimal(base_size = 12) +
theme(
  axis.ticks.x      = element_blank(),
  panel.grid.major.x = element_blank(),
  plot.title        = element_text(face = "bold", size = 14)
)

```

Claim Count Dispersion by Portfolio



Part 2: Demonstrating heterogeneity in portfolio 3

```

library(dplyr)
library(ggplot2)
library(MASS)

# 1. Prepare the data
cts3 <- df3 %>%
  group_by(id) %>%
  summarise(claims = ifelse(all(claim_size == 0), 0L, n())) %>%
  ungroup()

obs_df <- cts3 %>% count(claims) %>% rename(observed = n)

pois3 <- glm(claims ~ 1, family = poisson(), data = cts3)
nb3   <- glm.nb(claims ~ 1, data = cts3)

```

```

lam3 <- exp(coef(pois3))
mu3 <- exp(coef(nb3)); th3 <- nb3$theta

maxc <- max(cts3$claims)
model_df <- tibble(
  claims = 0:maxc,
  observed = obs_df$observed,
  Poisson = dpois(claims, lam3) * nrow(cts3),
  NegBin = dnbinom(claims, size = th3, mu = mu3) * nrow(cts3)
)

# 2. Pivot longer for ggplot
plot_df <- model_df %>%
  pivot_longer(cols = c("observed", "Poisson", "NegBin"),
               names_to = "Model", values_to = "Count")

# 3. Plot
ggplot(plot_df, aes(x = claims, y = Count, fill = Model, color = Model, shape = Model)) +
  # observed as bars
  geom_col(data = filter(plot_df, Model == "observed"),
            aes(x = claims, y = Count, fill = Model),
            color = NA, width = 0.6) +
  # Poisson & NegBin as lines + points
  geom_line(data = filter(plot_df, Model != "observed"),
            aes(x = claims, y = Count, color = Model), size = 1) +
  geom_point(data = filter(plot_df, Model != "observed"),
             aes(x = claims, y = Count, color = Model, shape = Model), size = 2) +
  # manual scales
  scale_fill_manual(
    name = "",
    values = c("observed" = "lightgray", "Poisson" = NA, "NegBin" = NA)
  ) +
  scale_color_manual(
    name = "",
    values = c("observed" = NA, "Poisson" = "#4C72B0", "NegBin" = "#C44E52")
  ) +
  scale_shape_manual(
    name = "",
    values = c("observed" = NA, "Poisson" = 16, "NegBin" = 17)
  ) +
  scale_x_continuous(breaks = 0:maxc) +
  labs(
    x = "Number of Claims per Policy",
    y = "Number of Policies",
    title = "Portfolio 3: Observed vs Poisson & NegBin Fits"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold"),
    panel.grid.major.x = element_blank(),
    axis.ticks.x = element_line(),
    axis.text.x = element_text(margin = margin(t = 5)),
    legend.position = "top"
  )

```

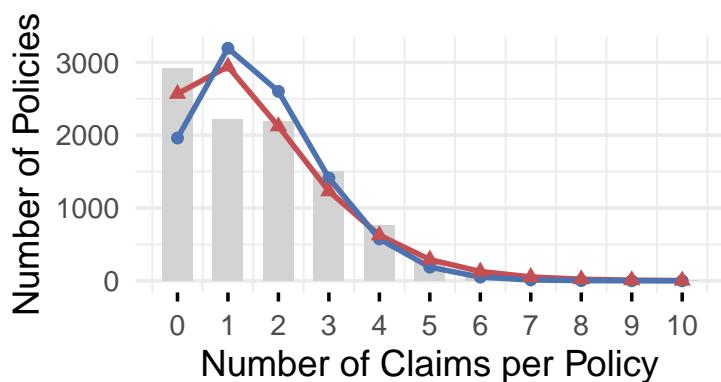
```

)
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

Portfolio 3: Observed vs Pois

· NegBin ● Poisson ▲ NegBin ■ observed



Question 3

```

library(readxl)
library(fitdistrplus)
library(actuar)    # for Pareto
library(extRemes)  # for mrlplot

## Loading required package: Lmoments

## Loading required package: distillery

##
## Attaching package: 'extRemes'

## The following objects are masked from 'package:stats':
## 
##     qqnorm, qqplot

# 1. Data
df1  <- read_excel("1463797.xlsx", sheet="Sheet1")
sev1 <- df1$claim_size[df1$claim_size > 0]
n <- length(sev1)

```

```

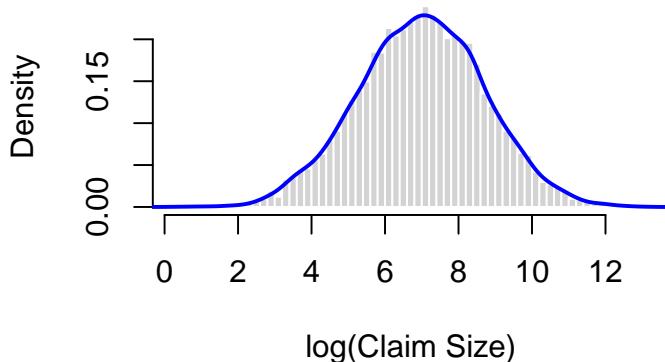
logs1 <- log(sev1)

# 2. Fit candidates (for Q-Q)
fit_gamma <- fitdist(sev1, "gamma", start=list(shape=1,scale=mean(sev1)))
fit_lnorm <- fitdist(sev1, "lnorm")
fit_pareto <- fitdist(sev1, "pareto", start=list(shape=2,scale=min(sev1)))

# Empirical density (logged data)
par(mar = c(5,4,4,2) + 0.1) # extra top margin for title
hist(logs1,
      breaks = 50,
      freq   = FALSE,
      col    = "lightgray",
      border = "white",
      xlab   = "log(Claim Size)",
      ylab   = "Density",
      main   = "P1: Log Scale Histogram & Density"
)
lines(density(logs1), col = "blue", lwd = 2)

```

P1: Log Scale Histogram & Density

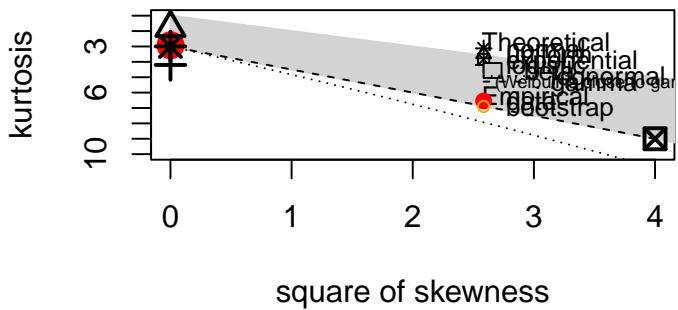


```

# 3a) Cullen-Frey on log-data
par(mar = c(5,4,6,2))      # bottom, left, top, right margins
descdist(logs1, boot=1000, discrete=FALSE)

```

Cullen and Frey graph

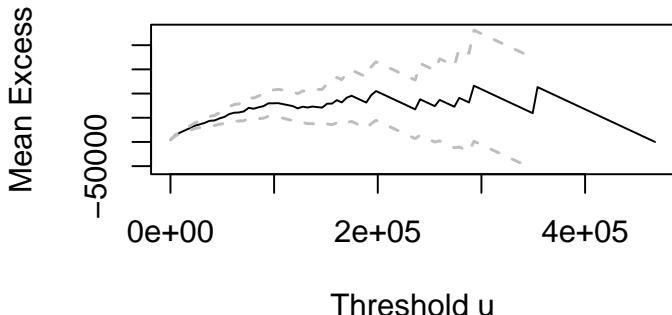


```
## summary statistics
## -----
## min: 0.3577616   max: 13.05511
## median: 7.012698
## mean: 6.989846
## estimated sd: 1.710654
## estimated skewness: -0.02848522
## estimated kurtosis: 2.923099

# Default title "Cullen and Frey graph" sits neatly in that top margin.

# 3b) Mean-excess plot
par(mar = c(5,4,6,2))
mrlplot(sev1,
       main = "P1: Mean-Excess Plot",
       xlab = "Threshold u")
```

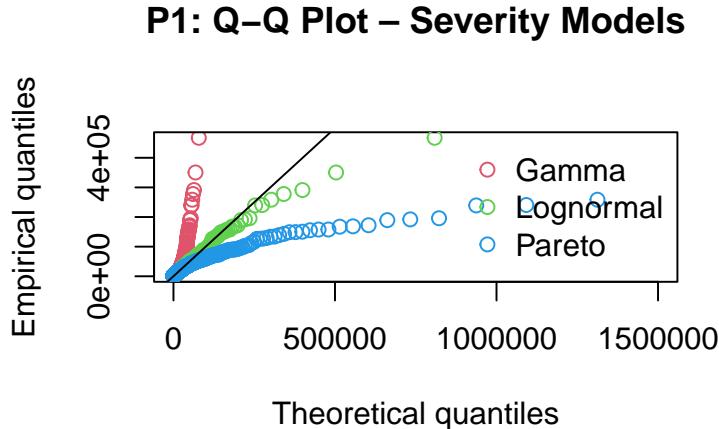
P1: Mean-Excess Plot



```

# 3c) Q-Q comparison
par(mar = c(5,4,6,2))
qqcomp(
  list(fit_gamma, fit_lnorm, fit_pareto),
  legendtext = c("Gamma","Lognormal","Pareto"),
  main = "P1: Q-Q Plot - Severity Models",
  xlab = "Theoretical quantiles",
  ylab = "Empirical quantiles",
  xlim = c(0, 1500000)
)

```



```

# 3d) AIC and BIC comparison
fits <- list(
  Gamma = fit_gamma,
  Lognormal = fit_lnorm,
  Pareto = fit_pareto
)

ic_tab <- t(sapply(fits, function(fit) {
  k <- length(fit$estimate)
  ll <- fit$loglik
  AIC <- 2*k - 2*ll
  BIC <- log(n)*k - 2*ll
  c(AIC = round(AIC,1), BIC = round(BIC,1))
} ))
print(ic_tab)

##          AIC      BIC
## Gamma    164796.8 164811.0
## Lognormal 161740.5 161754.8
## Pareto   161931.7 161945.9

```

```

# Estimating parameters of log normal distribution
fit_lnorm <- fitdist(sev1, "lnorm")
meanlog <- fit_lnorm$estimate["meanlog"]
sdlog   <- fit_lnorm$estimate["sdlog"]
meanlog; sdlog

## meanlog
## 6.989846

## sdlog
## 1.71056

library(ggplot2)

# Parameters
mu      <- 0
sigma   <- 1

# Data frames
norm_df <- data.frame(
  x = seq(-4, 8, length.out = 1000),
  y = dnorm(seq(-4, 8, length.out = 1000), mean = mu, sd = sigma),
  Dist = "Normal(0,1)"
)

log_df <- data.frame(
  x = seq(0.01, 8, length.out = 1000),
  y = dlnorm(seq(0.01, 8, length.out = 1000), meanlog = mu, sdlog = sigma),
  Dist = "Lognormal(0,1)"
)

# Combine
plot_df <- rbind(norm_df, log_df)

# Plot
ggplot(plot_df, aes(x = x, y = y, color = Dist)) +
  geom_line(size = 1) +
  scale_color_manual(
    values = c("Normal(0,1)" = "#2C3E50", "Lognormal(0,1)" = "#E67E22")
  ) +
  labs(
    title = "Normal vs Lognormal Densities",
    x     = "x",
    y     = "Density"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title    = element_text(face = "bold"),
    legend.position = c(0.02, 0.98),           # near top-left
    legend.justification = c(0, 1),            # left/top corner of legend at that point
    legend.background = element_rect(fill = "white", color = "gray80")
  )

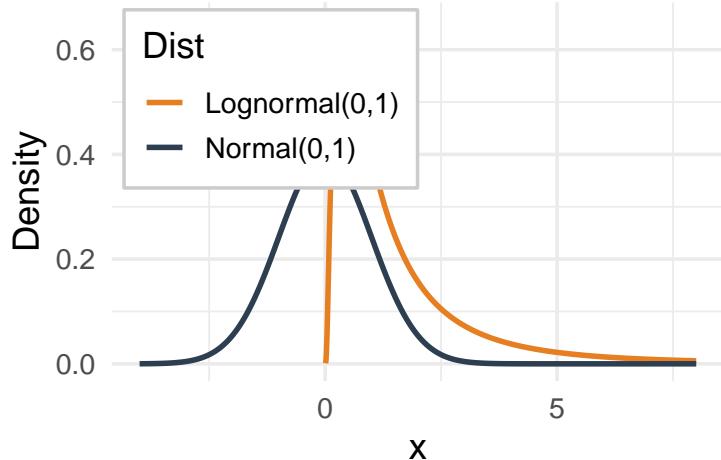
```

```

## Warning: A numeric 'legend.position' argument in 'theme()' was deprecated in ggplot2
## 3.5.0.
## i Please use the 'legend.position.inside' argument of 'theme()' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

Normal vs Lognormal Densities



Question 4

```

library(dplyr)

# Aggregate per-policy totals (example for Portfolio 1)
agg <- df2 %>%
  group_by(id) %>%
  summarise(total = sum(claim_size)) %>%
  ungroup()

n      <- nrow(agg)
p      <- mean(agg$total > 0)           # ~59%
meanN <- n * p                          # 5 900
varN  <- n * p * (1 - p)                # 2 419
p

## [1] 0.8308

tibble(
  Model      = c("Binomial", "Poisson"),
  Mean       = c(meanN,      meanN),
  Variance   = c(varN,      meanN),
  Dispersion = c(varN/meanN, 1)
)

```

```

## # A tibble: 2 x 4
##   Model      Mean Variance Dispersion
##   <chr>     <dbl>    <dbl>      <dbl>
## 1 Binomial  8308     1406.      0.169
## 2 Poisson   8308     8308       1

# Binomial Dispersion = 0.41 vs Poisson Dispersion = 1.0

library(fitdistrplus)

cts3 <- df3 %>%
  group_by(id) %>%
  summarise(
    # count how many non-zero claim_size entries each policy has
    claims = sum(claim_size > 0)
  ) %>%
  ungroup()

fit_nb3b <- fitdist(cts3$claims, "nbinom")

fit_nb3b$estimate

##      size        mu
## 3.860821 1.629247

```