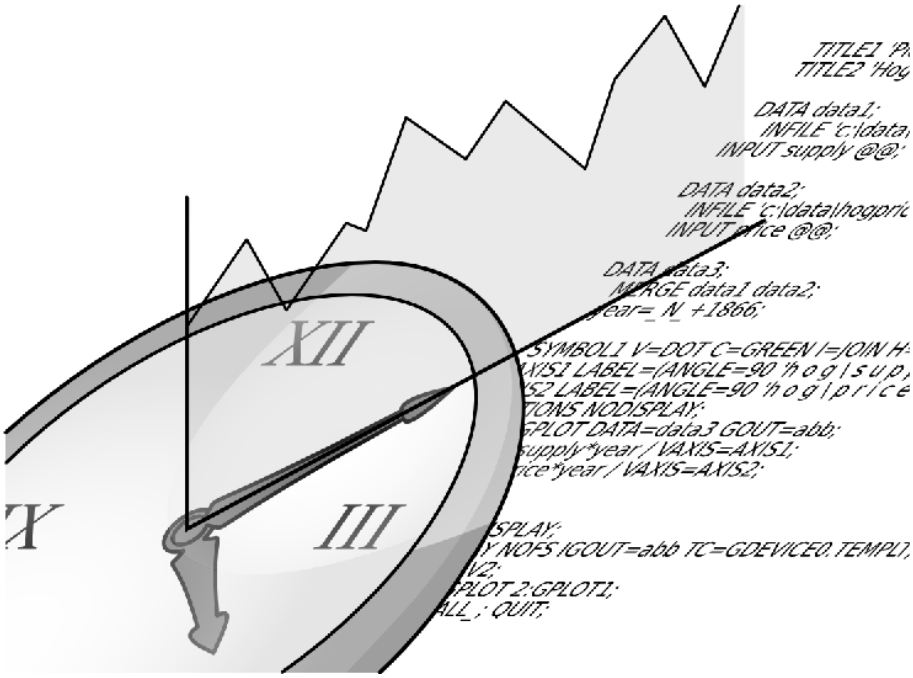
A First Course on Time Series Analysis

Examples with SAS



Chair of Statistics
University of Würzburg

Version: 2005.March.01
Copyright © 2005 Michael Falk.

Editors	Michael Falk, Frank Marohn, René Michel, Daniel Hofmann, Maria Macke
Programs	Bernward Tewes, René Michel, Daniel Hofmann

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

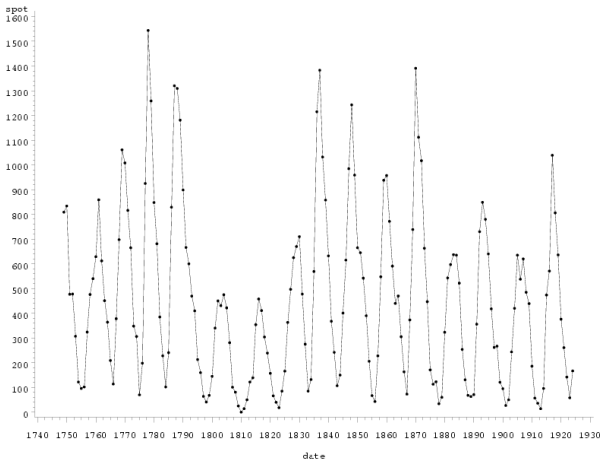
SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. Windows is a trademark, Microsoft is a registered trademark of the Microsoft Corporation.

The authors accept no responsibility for errors in the programs mentioned of their consequences.

Preface

The analysis of real data by means of statistical methods with the aid of a software package common in industry and administration usually is not an integral part of mathematics studies, but it will certainly be part of a future professional work.

The practical need for an investigation of time series data is exemplified by the following plot, which displays the yearly sunspot numbers between 1749 and 1924. These data are also known as the Wolf or Wölfer (a student of Wolf) Data. For a discussion of these data and further literature we refer to Wei (1990), Example 5.2.5.



The present book links up elements from time series analysis with a selection of statistical procedures used in general practice including the statistical software package SAS (Statistical Analysis System). Consequently this book addresses students of statistics as well as students of other branches such as economics, demography and engineering, where lectures on statistics belong to their academic training. But it is also intended for the practician who, beyond the use of statistical tools, is interested in their mathematical background. Numerous problems illustrate the applicability of the presented statistical procedures, where SAS gives the solutions. The programs used are explicitly listed and explained. No previous experience is expected neither in SAS nor in a special computer system so that a short training period is guaranteed.

This book is meant for a two semester course (lecture, seminar or practical training) where the first two chapters can be dealt with in the first semester. They

provide the principal components of the analysis of a time series in the time domain. Chapters 3, 4 and 5 deal with its analysis in the frequency domain and can be worked through in the second term. In order to understand the mathematical background some terms are useful such as convergence in distribution, stochastic convergence, maximum likelihood estimator as well as a basic knowledge of the test theory, so that work on the book can start after an introductory lecture on stochastics. Each chapter includes exercises. An exhaustive treatment is recommended.

Due to the vast field a selection of the subjects was necessary. Chapter 1 contains elements of an exploratory time series analysis, including the fit of models (logistic, Mitscherlich, Gompertz curve) to a series of data, linear filters for seasonal and trend adjustments (difference filters, Census $X-11$ Program) and exponential filters for monitoring a system. Autocovariances and autocorrelations as well as variance stabilizing techniques (Box–Cox transformations) are introduced. Chapter 2 provides an account of mathematical models of stationary sequences of random variables (white noise, moving averages, autoregressive processes, ARIMA models, cointegrated sequences, ARCH- and GARCH-processes, state-space models) together with their mathematical background (existence of stationary processes, covariance generating function, inverse and causal filters, stationarity condition, Yule–Walker equations, partial autocorrelation). The Box–Jenkins program for the specification of ARMA-models is discussed in detail (AIC, BIC and HQC information criterion). Gaussian processes and maximum likelihod estimation in Gaussian models are introduced as well as least squares estimators as a nonparametric alternative. The diagnostic check includes the Box–Ljung test. Many models of time series can be embedded in state-space models, which are introduced at the end of Chapter 2. The Kalman filter as a unified prediction technique closes the analysis of a time series in the time domain. The analysis of a series of data in the frequency domain starts in Chapter 3 (harmonic waves, Fourier frequencies, periodogram, Fourier transform and its inverse). The proof of the fact that the periodogram is the Fourier transform of the empirical autocovariance function is given. This links the analysis in the time domain with the analysis in the frequency domain. Chapter 4 gives an account of the analysis of the spectrum of the stationary process (spectral distribution function, spectral density, Herglotz's theorem). The effects of a linear filter are studied (transfer and power transfer function, low pass and high pass filters, filter design) and the spectral densities of ARMA-processes are computed. Some basic elements of a statistical analysis of a series of data in the frequency domain are provided in Chapter 5. The problem of testing for a white noise is dealt with (Fisher's κ-statistic, Bartlett–Kolmogorov–Smirnov test) together with the estimation of the spectral density (periodogram, discrete spectral average estimator, kernel estimator, confidence intervals).

This book is consecutively subdivided in a statistical part and an SAS-specific part. For better clearness the SAS-specific part, including the diagrams generated

with SAS, always starts with a computer symbol, representing the beginning of a session at the computer, and ends with a printer symbol for the end of this session.

This SAS-specific part is again divided in a diagram created with SAS, the program, which generated the diagram, and explanations to this program. In order to achieve a further differentiation between SAS-commands and individual nomenclature, SAS-specific commands were written in CAPITAL LETTERS, whereas individual notations were written in lower-case letters.

Contents

Chapter 1

Elements of Exploratory Time Series Analysis

A time series is a sequence of observations that are arranged according to the time of their outcome. The annual crop yield of sugar-beets and their price per ton for example is recorded in agriculture. The newspapers' business sections report daily stock prices, weekly interest rates, monthly rates of unemployment and annual turnovers. Meteorology records hourly wind speeds, daily maximum and minimum temperatures and annual rainfall. Geophysics is continuously observing the shaking or trembling of the earth in order to predict possibly impending earthquakes. An electroencephalogram traces brain waves made by an electroencephalograph in order to detect a cerebral disease, an electrocardiogram traces heart waves. The social sciences survey annual death and birth rates, the number of accidents in the home and various forms of criminal activities. Parameters in a manufacturing process are permanently monitored in order to carry out an on-line inspection in quality assurance.

There are, obviously, numerous reasons to record and to analyze the data of a time series. Among these is the wish to gain a better understanding of the data generating mechanism, the prediction of future values or the optimal control of a system. The characteristic property of a time series is the fact that the data are *not generated independently*, their dispersion varies in time, they are often governed by a trend and they have cyclic components. Statistical procedures that suppose *independent* and *identically distributed* data are, therefore, excluded from the analysis of time series. This requires proper methods that are summarized under *time series analysis*.

1.1 The Additive Model for a Time Series

The additive model for a given time series $y_1, \ldots, y_n$ is the assumption that these data are realizations of random variables Y_t that are themselves sums of four components

$$Y_t = T_t + Z_t + S_t + R_t, \qquad t = 1, \ldots, n. \tag{1.1}$$

where T_t is a (monotone) function of t, called *trend*, and Z_t reflects some nonrandom long term cyclic influence. Think of the famous business cycle usually consisting of recession, recovery, growth, and decline. S_t describes some nonrandom short term cyclic influence like a seasonal component whereas R_t is a random variable grasping all the deviations from the ideal non-stochastic model $y_t = T_t + Z_t + S_t$. The variables T_t and Z_t are often summarized as

$$G_t = T_t + Z_t, \tag{1.2}$$

describing the long term behavior of the time series. We suppose in the following that the expectation $\mathrm{E}(R_t)$ of the error variable exists and equals zero, reflecting the assumption that the random deviations above or below the nonrandom model balance each other on the average. Note that $\mathrm{E}(R_t) = 0$ can always be achieved by appropriately modifying one or more of the nonrandom components.

Example 1.1.1. (Unemployed1 Data). The following data y_t, $t = 1, \ldots, 51$, are the monthly numbers of unemployed workers in the building trade in Germany from July 1975 to September 1979.

MONTH	T	UNEMPLYD
July	1	60572
August	2	52461
September	3	47357
October	4	48320
November	5	60219
December	6	84418
January	7	119916
February	8	124350
March	9	87309
April	10	57035
May	11	39903
June	12	34053
July	13	29905

August	14	28068
September	15	26634
October	16	29259
November	17	38942
December	18	65036
January	19	110728
February	20	108931
March	21	71517
April	22	54428
May	23	42911
June	24	37123
July	25	33044
August	26	30755
September	27	28742
October	28	31968
November	29	41427
December	30	63685
January	31	99189
February	32	104240
March	33	75304
April	34	43622
May	35	33990
June	36	26819
July	37	25291
August	38	24538
September	39	22685
October	40	23945
November	41	28245
December	42	47017
January	43	90920
February	44	89340
March	45	47792
April	46	28448
May	47	19139
June	48	16728
July	49	16523
August	50	16622
September	51	15499

Figure 1.1.1. Listing of Unemployed1 Data.

```
***     Program 1_1_1   ***;
TITLE1 'Listing';
TITLE2 'Unemployed1 Data';

DATA data1;
```

```
 INFILE 'c:\data\unemployed1.txt';
 INPUT month $ t unemplyd;

PROC PRINT DATA = data1 NOOBS;
RUN;QUIT;
```

This program consists of two main parts, a `DATA` and a `PROC` step.

The `DATA` step started with the `DATA` statement creates a temporary dataset named `data1`. The purpose of `INFILE` is to link the `DATA` step to a raw dataset outside the program. The pathname of this dataset depends on the operating system; we will use the syntax of MS-DOS, which is most commonly known. `INPUT` tells SAS how to read the data. Three variables are defined here, where the first one contains character values. This is determined by the `$` sign behind the variable name. For each variable one value per line is read from the source into the computer's memory.

The statement `PROC procedurename DATA=filename;` invokes a procedure that is linked to the data from `filename`. Without the option `DATA=filename` the most recently created file is used.

The `PRINT` procedure lists the data; it comes with numerous options that allow control of the variables to be printed out, 'dress up' of the display etc. The SAS internal observation number (`OBS`) is printed by default, `NOOBS` suppresses the column of observation numbers on each line of output. An optional `VAR` statement determines the order (from left to right) in which variables are displayed. If not specified (like here), all variables in the data set will be printed in the order they were defined to SAS. Entering `RUN;` at any point of the program tells SAS that a unit of work (`DATA` step or `PROC`) ended. SAS then stops reading the program and begins to execute the unit. The `QUIT;` statement at the end terminates the processing of SAS.

A line starting with an asterisk `*` and ending with a semicolon `;` is ignored. These comment statements may occur at any point of the program except within raw data or another statement.

The `TITLE` statement generates a title. Its printing is actually suppressed here and in the following.

The following plot of the Unemployed1 Data shows a seasonal component and a downward trend. The period from July 1975 to September 1979 might be too short to indicate a possibly underlying long term business cycle.

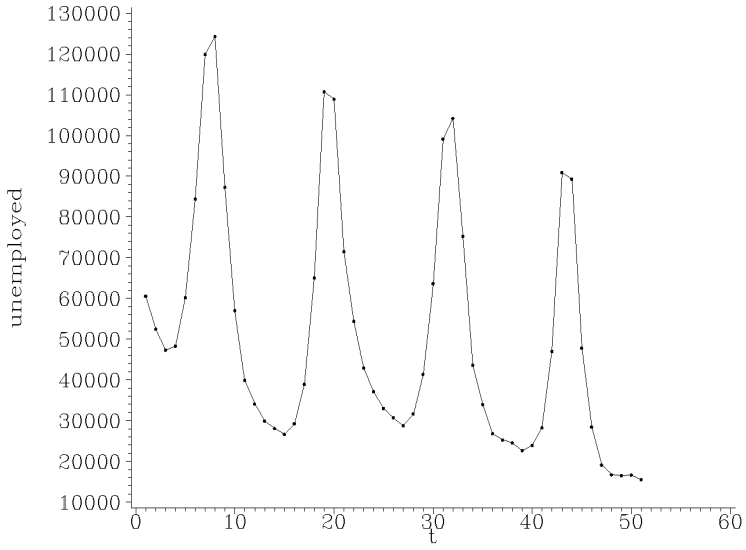


Figure 1.1.2. Plot of Unemployed1 Data.

```
***     Program 1_1_2   ***;
TITLE1 'Plot';
TITLE2 'Unemployed1 Data';

DATA data1;
INFILE 'c:\data\unemployed1.txt';
INPUT month $ t unemplyd;

AXIS1 LABEL=(ANGLE=90 'unemployed');
AXIS2 LABEL=('t');
SYMBOL1 V=DOT C=GREEN I=JOIN H=0.4 W=1;
PROC GPLOT DATA=data1;
    PLOT unemplyd*t / VAXIS=AXIS1 HAXIS=AXIS2;
RUN; QUIT;
```

Variables can be plotted by using the `GPLOT` procedure, where the graphical output is controlled by numerous options.

The `AXIS` statements with the `LABEL` options control labelling of the vertical and horizontal axes. `ANGLE=90` causes a rotation of the label of 90° so that it parallels the (vertical) axis in this example.

The `SYMBOL` statement defines the manner in which the data are displayed. `V=DOT`

`C=GREEN I=JOIN H=0.4 W=1` tell SAS to plot green dots of height 0.4 and to join them with a line of width 1. The `PLOT` statement in the `GPLOT` procedure is of the form `PLOT y-variable*x-variable / options;`, where the options here define the horizontal and the vertical axes.

Models with a Nonlinear Trend

In the additive model $Y_t = T_t + R_t$, where the nonstochastic component is only the trend T_t reflecting the growth of a system, and assuming $\mathrm{E}(R_t) = 0$, we have

$$\mathrm{E}(Y_t) = T_t =: f(t).$$

A common assumption is that the function f depends on several (unknown) *parameters* $\beta_1, \ldots, \beta_p$, i.e.,

$$f(t) = f(t; \beta_1, \ldots, \beta_p). \tag{1.3}$$

However, the *type* of the function f is known. The parameters $\beta_1, \ldots, \beta_p$ are then to be estimated from the set of realizations y_t of the random variables Y_t. A common approach is a *least squares estimate* $\hat{\beta}_1, \ldots, \hat{\beta}_p$ satisfying

$$\sum_t \Big(y_t - f(t; \hat{\beta}_1, \ldots, \hat{\beta}_p)\Big)^2 = \min_{\beta_1,\ldots,\beta_p} \sum_t \Big(y_t - f(t; \beta_1, \ldots, \beta_p)\Big)^2, \tag{1.4}$$

whose computation, if it exists at all, is a numerical problem. The value $\hat{y}_t := f(t; \hat{\beta}_1, \ldots, \hat{\beta}_p)$ can serve as a *prediction* of a future y_t. The *observed* differences $y_t - \hat{y}_t$ are called *residuals*. They contain information about the goodness of the fit of our model to the data. In the following we list several popular examples of trend functions.

The Logistic Function

The function

$$f_{\log}(t) := f_{\log}(t; \beta_1, \beta_2, \beta_3) := \frac{\beta_3}{1 + \beta_2 \exp(-\beta_1 t)}, \qquad t \in \mathbb{R}, \tag{1.5}$$

with $\beta_1, \beta_2, \beta_3 \in \mathbb{R} \setminus \{0\}$ is the widely used *logistic function*.

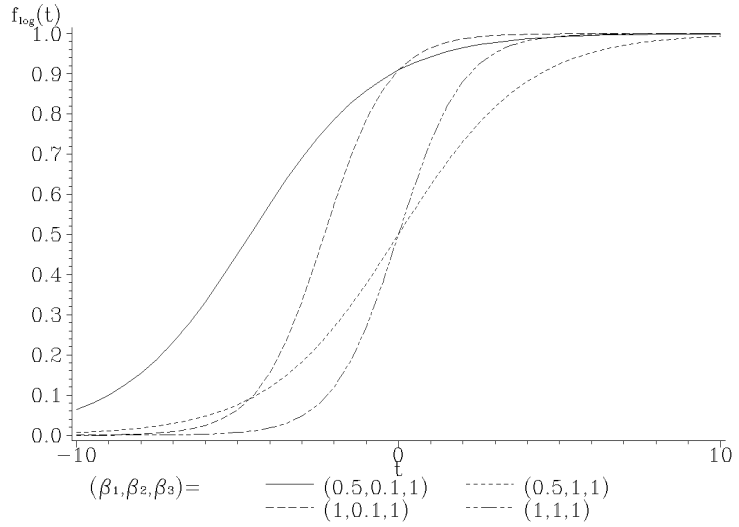


Figure 1.1.3. The logistic function $f_{\log}$ with different values of $\beta_1, \beta_2, \beta_3$.

```
***     Program 1_1_3   ***;
TITLE1 'Plots of the Logistic Function';

DATA data1;
 beta3=1;
 DO beta1= 0.5, 1;
   DO beta2=0.1, 1;
     DO t=-10 TO 10 BY 0.5;
        s = COMPRESS('(' || beta1 || ',' || beta2 ||
                     ',' || beta3 || ')');
        f_log=beta3/(1+beta2*EXP(-beta1*t));
    OUTPUT;
   END; END; END;


SYMBOL1 C=GREEN V=NONE I=JOIN L=1;
SYMBOL2 C=GREEN V=NONE I=JOIN L=2;
SYMBOL3 C=GREEN V=NONE I=JOIN L=3;
SYMBOL4 C=GREEN V=NONE I=JOIN L=33;
AXIS1 LABEL=(H=2 'f' H=1 'log' H=2 '(t)');
AXIS2 LABEL=('t');
```

```
LEGEND1 LABEL=(F=CGREEK H=2 '(b' H=1 '1' H=2 ',
         b' H=1 '2' H=2 ',b' H=1 '3' H=2 ')=');
PROC GPLOT DATA=data1;
    PLOT f_log*t=s / VAXIS=AXIS1 HAXIS=AXIS2
        LEGEND=LEGEND1;
RUN; QUIT;
```

A function is plotted by computing its values at numerous grid points and then joining them. The computation is done in the `DATA` step, where the data file `data1` is generated. It contains the values of `f_log`, computed at the grid $t = -10, -9.5, \ldots, 10$ and indexed by the vector `s` of the different choices of parameters. This is done by nested `DO` loops. The operator `||` merges two strings and `COMPRESS` removes the empty space in the string. `OUTPUT` then stores the values of interest of `f_log`, `t` and `s` (and the other variables) in the data set `data1`. The four functions are plotted by the `GPLOT` procedure by adding `=s` in the `PLOT` statement. This also automatically generates a legend, which is customized by the `LEGEND1` statement. Here the label is modified by using a greek font (`F=CGREEK`) and generating smaller letters of height 1 for the indices, while assuming a normal height of 2 (`H=1` and `H=2`). The last feature is also used in the axis statement. For each value of `s` SAS takes a new `SYMBOL` statement. They generate lines of different line types (`L=1,2, 3, 33`).

We obviously have $\lim_{t\to\infty} f_{\log}(t) = \beta_3$, if $\beta_1 > 0$. The value β_3 often resembles the maximum impregnation or growth of a system. Note that

$$\begin{aligned}
\frac{1}{f_{\log}(t)} &= \frac{1+\beta_2\exp(-\beta_1 t)}{\beta_3} \\
&= \frac{1-\exp(-\beta_1)}{\beta_3} + \exp(-\beta_1)\frac{1+\beta_2\exp(-\beta_1(t-1))}{\beta_3} \\
&= \frac{1-\exp(-\beta_1)}{\beta_3} + \exp(-\beta_1)\frac{1}{f_{\log}(t-1)} \\
&= a + \frac{b}{f_{\log}(t-1)}. \qquad (1.6)
\end{aligned}$$

This means that there is a linear relationship among $1/f_{\log}(t)$. This can serve as a basis for estimating the parameters $\beta_1, \beta_2, \beta_3$ by an appropriate linear least squares approach, see Exercises 2 and 3. In the following example we fit the logistic trend model (1.5) to the population growth of the area of North Rhine-Westphalia (NRW), which is a federal state of Germany.

Example 1.1.2. (Population1 Data). The following table shows the population sizes y_t in millions of the area of North-Rhine-Westphalia in 5 years steps from 1935 to 1980 as well as their predicted values $\hat{y}_t$, obtained from a least squares estimation as described in (1.4) for a logistic model.

Year	t	Population sizes y_t (in millions)	Predicted values $\hat{y}_t$ (in millions)
1935	1	11.772	10.930
1940	2	12.059	11.827
1945	3	11.200	12.709
1950	4	12.926	13.565
1955	5	14.442	14.384
1960	6	15.694	15.158
1965	7	16.661	15.881
1970	8	16.914	16.548
1975	9	17.176	17.158
1980	10	17.044	17.710

Table 1.1.1. Population1 Data.

As a prediction of the population size at time t we obtain in the logistic model

$$\begin{aligned}\hat{y}_t &:= \frac{\hat{\beta}_3}{1+\hat{\beta}_2\exp(-\hat{\beta}_1 t)}\\ &= \frac{21.5016}{1+1.1436\exp(-0.1675\,t)}\end{aligned}$$

with the estimated saturation size $\hat{\beta}_3 = 21.5016$. The following plot shows the data and the fitted logistic curve.

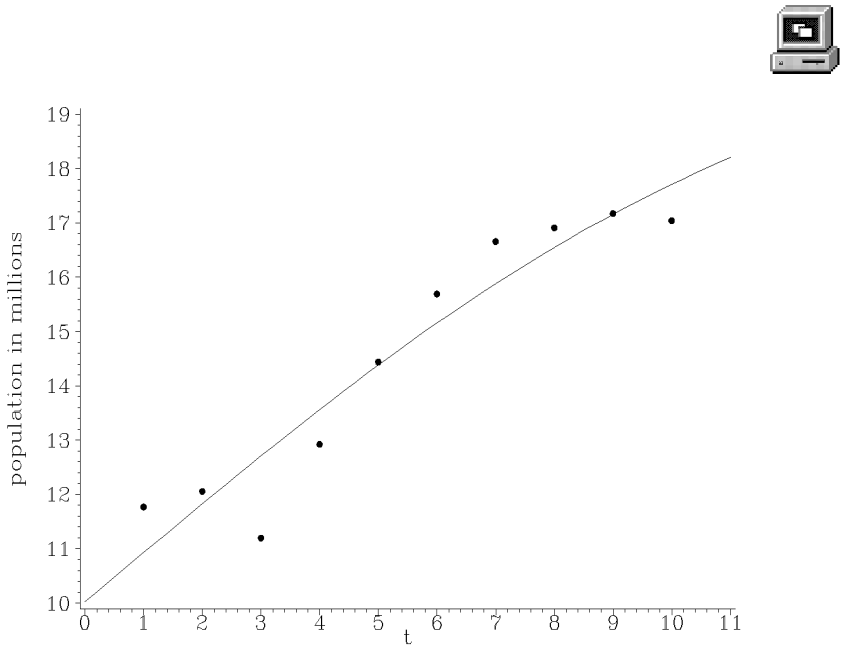


Figure 1.1.4. NRW population sizes and fitted logistic function.

```
***     Program 1_1_4   ***;
TITLE1 'Population sizes and logistic fit';
TITLE2 'Population1 Data';

DATA data1;
INFILE 'c:\data\population1.txt';
INPUT year t pop;

PROC NLIN DATA=data1 OUTEST=estimate;
    MODEL pop=beta3/(1+beta2*EXP(-beta1*t));
    PARAMETERS beta1=1 beta2=1 beta3=20;
RUN;

DATA data2;
SET estimate(WHERE=(_TYPE_='FINAL'));
DO t1=0 TO 11 BY 0.2;
  f_log=beta3/(1+beta2*EXP(-beta1*t1));
  OUTPUT;
END;

DATA data3;
```

```
    MERGE data1 data2;

AXIS1 LABEL=(ANGLE=90 'population in millions');
AXIS2 LABEL=('t');
SYMBOL1 V=DOT C=GREEN I=NONE;
SYMBOL2 V=NONE C=GREEN I=JOIN W=1;
PROC GPLOT DATA=data3;
    PLOT pop*t=1 f_log*t1=2 / OVERLAY VAXIS=AXIS1
                 HAXIS=AXIS2;
RUN; QUIT;
```

The procedure `NLIN` fits nonlinear regression models by least squares. The `OUTEST` option names the data set to contain the parameter estimates produced by `NLIN`. The `MODEL` statement defines the prediction equation by declaring the dependent variable and defining an expression that evaluates predicted values. A `PARAMETERS` statement must follow the `PROC NLIN` statement. Each `parameter=value` expression specifies the starting values of the parameter. Using the final estimates of `PROC NLIN` by the `SET` statement in combination with the `WHERE` data set option, the second data step generates the fitted logistic function values. The options in the `GPLOT` statement cause the data points and the predicted function to be shown in one plot, after they were stored together in a new data set `data3` merging `data1` and `data2` with the `MERGE` statement.

The Mitscherlich Function

The *Mitscherlich function* is typically used for modelling the long term growth of a system:

$$f_M(t) := f_M(t; \beta_1, \beta_2, \beta_3) := \beta_1 + \beta_2 \exp(\beta_3 t), \qquad t \geq 0, \tag{1.7}$$

where $\beta_1, \beta_2 \in \mathbb{R}$ and $\beta_3 < 0$. Since β_3 is negative we have $\lim_{t\to\infty} f_M(t) = \beta_1$ and thus the parameter β_1 is the saturation value of the system. The (initial) value of the system at the time $t = 0$ is $f_M(0) = \beta_1 + \beta_2$.

The Gompertz Curve

A further quite common function for modelling the increase or decrease of a system is the *Gompertz curve*

$$f_G(t) := f_G(t; \beta_1, \beta_2, \beta_3) := \exp(\beta_1 + \beta_2 \beta_3^t), \qquad t \geq 0, \tag{1.8}$$

where $\beta_1, \beta_2 \in \mathbb{R}$ and $\beta_3 \in (0, 1)$.

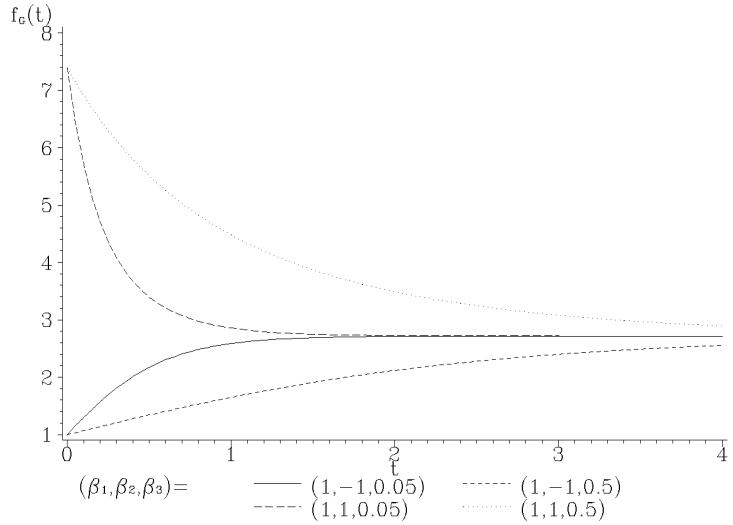


Figure 1.1.5. Gompertz curves with different parameters.

```
***     Program 1_1_5   ***;
TITLE1 'Gompertz curves';

DATA data1;
 DO beta1=1;
   DO beta2=-1, 1;
     DO beta3=0.05, 0.5;
       DO t=0 TO 4 BY 0.05;
         s = COMPRESS('(' || beta1 || ',' || beta2 ||
                          ',' || beta3 || ')');
         f_g=EXP(beta1+beta2*beta3**t);
     OUTPUT;
   END; END; END; END;

SYMBOL1 C=GREEN V=NONE I=JOIN L=1;
SYMBOL2 C=GREEN V=NONE I=JOIN L=2;
SYMBOL3 C=GREEN V=NONE I=JOIN L=3;
SYMBOL4 C=GREEN V=NONE I=JOIN L=33;
AXIS1 LABEL=(H=2 'f' H=1 'G' H=2 '(t)');
```