R Scripts for Longitudinal and Panel Data

Yumo Dong, Edward Frees, and others??

Contents

Preface

Date: 14 September 2019

Here are `R` scripts for the book **Longitudinal and Panel Data** by Edward W. Frees. See the book web site.

The datasets may be downloaded from downloaded from website.

We will review these scripts in our Panel and Copula Reading Group.

As a group, it may be worth our time to update and polish these scripts. They were first done in 2003 and have not received a lot of cleansing since that time. If you contribute, then this will help polish your `R` skills, as well as learn a bit about `Github`, where the scripts and this output is being hosted. For more on actuarial education on the web through `Github`, see the Open Actuarial Textbooks project.

Chapter 1

Introduction

1.1 Import Data

First, we can import "Divorce.txt" downloaded from website https://instruction.bus.wisc.edu/jfrees/jfreesbooks/Longitudinal%20and%20Panel%20Data/Book/DataFiles.htm

These are data describing the divorce rate in each state. In addition, there is other socioeconomic information about a state that may be related to the divorce rate. In particular, data concerning the number of marriages and births, unemployment and crime rates, and AFDC (Aid to Families with Dependent Children) payments are available. In this file, data are available for the years 1965, 1975, 1985 and 1995. The information provided by this study is potentially useful for governing agencies in budgeting for social needs such as judicial and welfare services that are affected by divorce. The data for the study were collected from various U.S. Statistical Abstracts. Divorce rate is defined as the number of divorces and annulments per thousand population per state. The independent variables include the number of marriages and live births per thousand population, the total unemployment rate as percent of total work force, the average monthly AFDC payments per family, and the total number of criminal offenses known to the police (murder, rape, robbery, aggravated assault, burglary, larceny, and motor vehicle theft). Some of the data points contain missing observations due to unavailability, and Nevada is unusual due to its uniquely high and unrepresentative marriage and divorce rates. Source: U.S. Statistical Abstract, various issues.

Variable	Description
DIVORCE	Number of divorces and annulments per state per one thousand population.
BIRTH	Number of live births per state per one thousand population.

Variable	Description
MARRIAGE	Number of marriages per state per one thousand population.
UNEMPLOY	Total unemployment rate as a percentage of the total work force.
CRIME	Total number of criminal offenses (murder, rape, robbery, aggravated assault, burglary, larceny and motor vehicle theft) known to police per one hundred thousand population.
AFDC	Average monthly AFDC (Aid to Families with Dependent Children) payments per family.
STATE	State identifier, 1-51.
TIME	Time identifier, 1-4.

```r
#  "\t"  INDICATES SEPARATED BY TABLES  ;
divorce = read.table("TXTData/Divorce.txt", sep ="\t", quote = "",header=TRUE)
# divorce = read.table(choose.files(), sep ="\t", quote = "",header=TRUE)
```

Let's have a look at the dataset. The names of variables and the first 8 rows observations.

```r
#  PROVIDES THE NAMES IN THE FILE AND LISTS THE FIRST 8 OBSERVATIONS  ;
names (divorce)
```

```
 [1] "DIVORCE"    "BIRTH"      "MARRIAGE"   "UNEMPLOY"   "CRIME"
 [6] "AFDC"       "STATE"      "TIME"       "STATE.Name" "Region"
```

```r
divorce[1:8,]
```

```
  DIVORCE BIRTH MARRIAGE UNEMPLOY  CRIME AFDC STATE TIME    STATE.Name
1     2.6  19.9      8.8      4.9  6.799  114     1    1         Maine
2     2.3  19.5     13.4      2.8  6.106  188     2    1 New Hampshire
3     1.5  20.5      9.0      4.2  5.793  113     3    1       Vermont
4     1.5  18.8      7.1      4.9 15.072  188     4    1 Massachusetts
5     1.3  19.4      7.1      4.9 14.180  172     5    1  Rhode Island
6     1.3  19.2      7.4      3.9 11.749  197     6    1   Connecticut
7     0.5  18.6      7.4      4.6 22.509  218     7    1      New York
8     0.8  18.5      6.8      5.1 13.966  203     8    1    New Jersey
           Region
1     New England
2     New England
3     New England
4     New England
5     New England
6     New England
7 Middle Atlantic
8 Middle Atlantic
```

We can check some summary statistics. The dimension of `divorce`.

```r
#  SUMMARY STATISTICS  ;
dim(divorce)
```

```
[1] 204  10
```

A summary of variables `DIVORCE` and `AFDC`.

```r
summary(divorce[, c("DIVORCE", "AFDC")])
```

```
    DIVORCE           AFDC      
 Min.   :0.500   Min.   : 33.0  
 1st Qu.:3.300   1st Qu.:154.0  
 Median :4.250   Median :224.0  
 Mean   :4.361   Mean   :245.9  
 3rd Qu.:5.300   3rd Qu.:315.0  
 Max.   :9.100   Max.   :731.0  
 NA's   :12      NA's   :3      
```

```r
sd(divorce[,c("DIVORCE")], na.rm=TRUE) #The standard deviation of DIVORCE.
```

```
[1] 1.704068
```

```r
sd(divorce[,c("AFDC")], na.rm=TRUE) #The standard deviation of AFDC.
```

```
[1] 122.2453
```

```r
cor(divorce$DIVORCE, divorce$AFDC, use="pairwise.complete.obs")# The correlation between DIVORCE
```

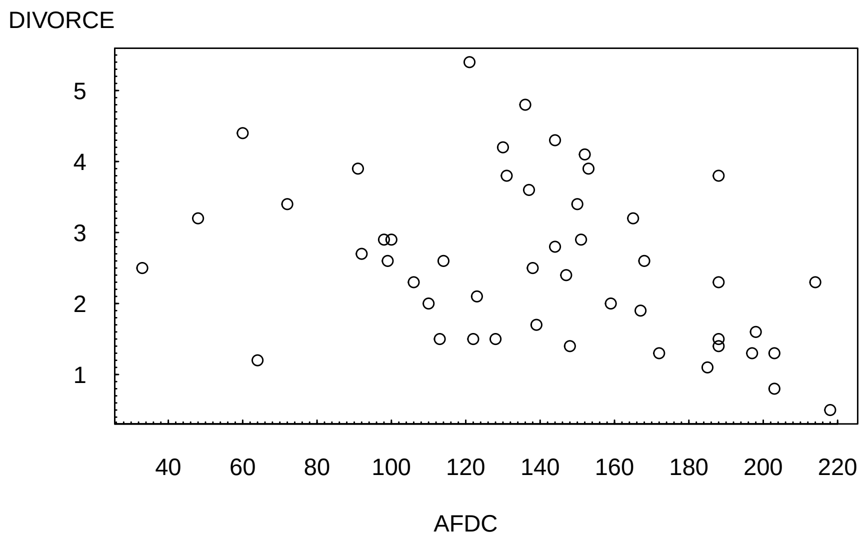
```
[1] 0.07306962
```

1.2 Example 1.1: Divorce Rates (page 2)

1.2.1 Figure 1.1: Plot of 1965 divorce rates versus AFDC payments.

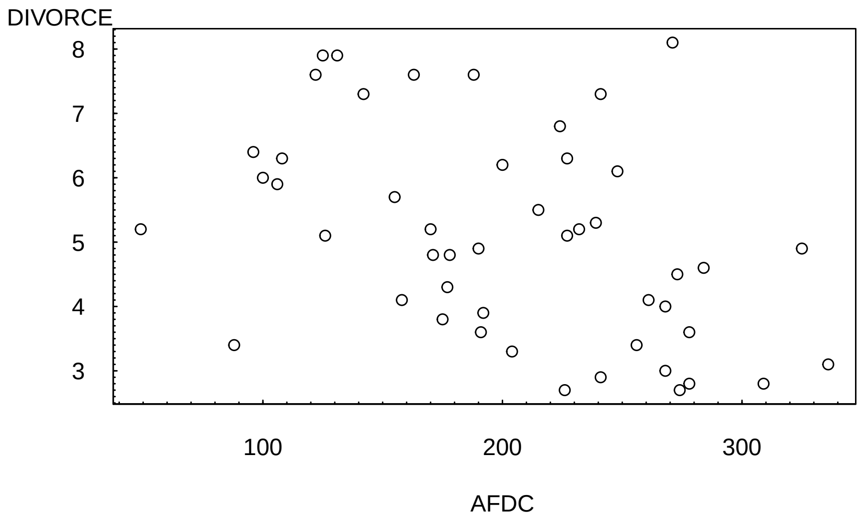
Figure 1.1 shows the 1965 divorce rates versus AFDC (Aid to Families with Dependent Children) payments for the fifty states.

```r
#  FIGURE 1.1. PLOT 1965 DATA ;
plot(DIVORCE ~ AFDC, subset=TIME %in% c(1),data = divorce, xaxt="n", yaxt="n",ylab="",xlab="")

axis(2, at=seq(0, 6, by=1), las=1, font=10, cex=0.005, tck=0.01)

axis(2, at=seq(0, 6, by=0.1), lab=F, tck=0.005)
axis(1, at=seq(20,220, by=20), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(20,220, by=2), lab=F, tck=0.005)
mtext("DIVORCE", side=2, line=0, at=6, font=12, cex=1, las=1)
mtext("AFDC", side=1, line=3, at=120, font=12, cex=1)
```



We can also plot 1975 data following the same method.

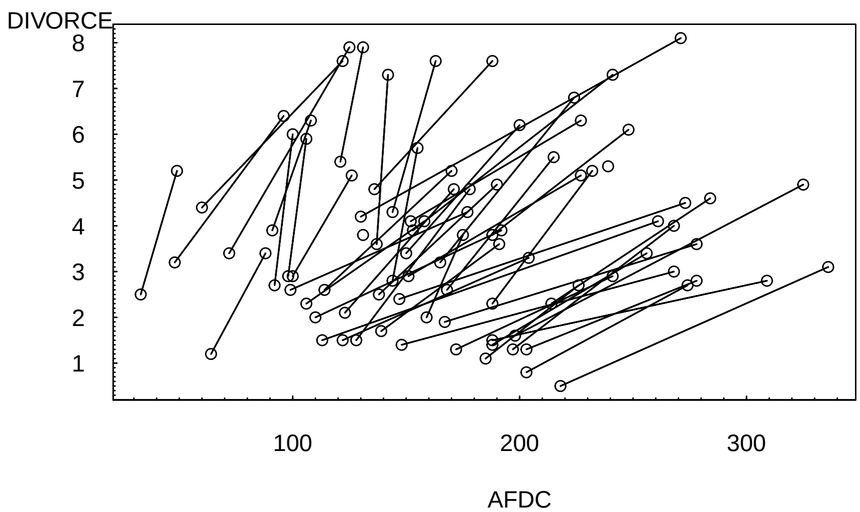
```
#  PLOT 1975 DATA ;
plot(DIVORCE ~ AFDC, subset=TIME %in% c(2),data = divorce,xaxt="n", yaxt="n",ylab="",xl
axis(2, at=seq(2, 9, by=1), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(2, 9, by=0.1), lab=F, tck=0.005)
axis(1, at=seq(0,400, by=100), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(0,400, by=10), lab=F, tck=0.005)
mtext("DIVORCE", side=2, line=0, at=8.5, font=12, cex=1, las=1)
mtext("AFDC", side=1, line=3, at=200, font=12, cex=1)
```



1.2.2 Figure 1.2: Plot of divorce rate versus AFDC payments from 1965 and 1975.

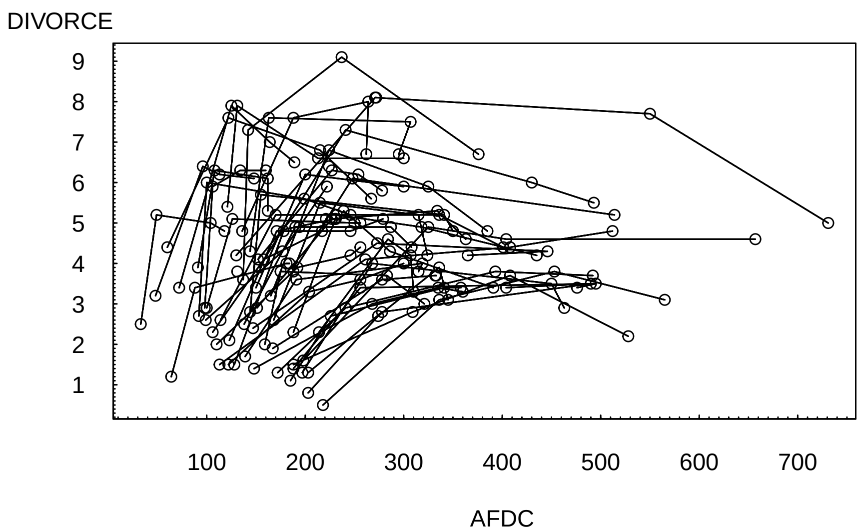
Figure 1.2 shows both the 1965 and 1975 data; a line connects the two observations within each state. These lines represent a change over time (dynamic), not a cross-sectional relationship.

```
plot(DIVORCE ~ AFDC, data = subset(divorce, TIME %in% c(1, 2)), xaxt="n", yaxt="n",ylab="",xlab='
   for (i in divorce$STATE) {
    lines(DIVORCE ~ AFDC, data = subset(divorce, TIME %in% c(1, 2) & STATE == i)) }
axis(2, at=seq(0, 10, by=1), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(0, 10, by=0.1), lab=F, tck=0.005)
axis(1, at=seq(0,400, by=100), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(0,400, by=10), lab=F, tck=0.005)
mtext("DIVORCE", side=2, line=0, at=8.5, font=12, cex=1, las=1)
mtext("AFDC", side=1, line=3, at=200, font=12, cex=1)
```



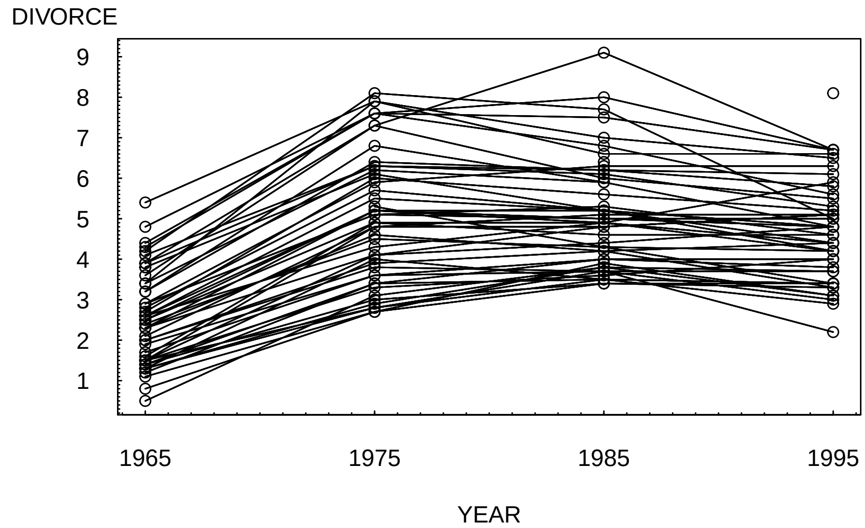
We can plot data for all years and connect the years.

```
#  PLOT ALL DATA, CONNECTING THE YEARS ;
plot(DIVORCE ~ AFDC, data = divorce, xaxt="n", yaxt="n",ylab="",xlab="")
  for (i in divorce$STATE) {
  lines(DIVORCE ~ AFDC, data = subset(divorce, STATE == i)) }
axis(2, at=seq(0, 10, by=1), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(0, 10, by=0.1), lab=F, tck=0.005)
axis(1, at=seq(0,800, by=100), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(0,800, by=10), lab=F, tck=0.005)
mtext("DIVORCE", side=2, line=0, at=10, font=12, cex=1, las=1)
mtext("AFDC", side=1, line=3, at=400, font=12, cex=1)
```



We can also look at the multiple time series plot by the `STATE`.

```
#  MULTIPLE TIME SERIES PLOT  ;
divorce$YEAR=divorce$TIME*10+1955
plot(DIVORCE ~ YEAR, data = divorce, xaxt="n", yaxt="n",ylab="",xlab="")
   for (i in divorce$STATE) {
   lines(DIVORCE ~ YEAR, data = subset(divorce, STATE == i)) }
axis(2, at=seq(0, 10, by=1), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(0, 10, by=0.1), lab=F, tck=0.005)
axis(1, at=seq(1965,1995, by=10), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(1964,2000, by=1), lab=F, tck=0.005)
mtext("DIVORCE", side=2, line=0, at=10, font=12, cex=1, las=1)
mtext("YEAR", side=1, line=3, at=1980, font=12, cex=1)
```


DIVORCE
9
8
7
6
5
4
3
2
1
1965
1975
1985
1995
YEAR

Chapter 2

Fixed-Effects Models

2.1 Import Data

We consider T=6 years, 1990-1995, of data for inpatient hospital charges that are covered by the Medicare program. The data were obtained from the Health Care Financing Administration, Bureau of Data Management and Strategy. To illustrate, in 1995 the total covered charges were $157.8 billions for twelve million discharges. For this analysis, we use state as the subject, or risk class. Thus, we consider n=54 states that include the 50 states in the Union, the District of Columbia, Virgin Islands, Puerto Rico and an unspecified "other" category.

Variable	Description
STATE	State identifier, 1-54
YEAR	Year identifier, 1-6
TOT_CHG	Total hospital charges, in millions of dollars.
COV_CHG	Total hospital charges covered by Medicare, in millions of dollars.
MED_REIM	Total hospital charges reimbursed by the Medicare program, in millions of dollars.
TOT_D	Total number of hospitals stays, in days.
NUM_DSHG	Number discharged, in thousands.
AVE_T_D	Average hospital stay per discharge in days.

```
#  "\t"  INDICATES SEPARATED BY TABLES  ;
Medicare  = read.table("TXTData/Medicare.txt", sep ="\t", quote = "",header=TRUE)

# Medicare = read.table(choose.files(), sep ="\t", quote = "",header=TRUE)
```

Let's have a look at the dataset. The names of variables and the first 8 rows observations.

```
#  PROVIDES THE NAMES IN THE FILE AND LISTS THE FIRST 8 OBSERVATIONS   ;
names (Medicare)
```

```
[1] "STATE"    "YEAR"     "TOT_CHG"  "COV_CHG"  "MED_REIB" "TOT_D"
[7] "NUM_DCHG" "AVE_T_D"  "NMSTATE"
```

```
Medicare [1:8, ]
```

```
  STATE YEAR    TOT_CHG    COV_CHG   MED_REIB   TOT_D NUM_DCHG AVE_T_D
1     1    1 2211617271 2170240349  972752944 1932673   230015       8
2     1    2 2523987347 2468263759 1046016144 1936939   234739       8
3     1    3 2975969979 2922611694 1205791592 2016354   245027       8
4     1    4 3194595003 3149745611 1307982985 1948427   243947       8
5     1    5 3417704863 3384305357 1376211788 1926335   258384       7
6     1    6 3519375275 3492635576 1466220936 1847216   261738       7
7     2    1   64747759   62242279   42083051   51923     6636       8
8     2    2   70600503   67579913   46928596   53051     6940       8
  NMSTATE
1      AL
2      AL
3      AL
4      AL
5      AL
6      AL
7      AK
8      AK
```

Then we need to create some other variables for later use.

```
#  CREATE OTHER VARIABLES;
# Firstly, we need change the names of existing variables.
names(Medicare)[names(Medicare)=="TOT_CHG"]="TOT.CHG";
names(Medicare)[names(Medicare)=="COV_CHG"]="COV.CHG";
names(Medicare)[names(Medicare)=="MED_REIB"]="MED.REIB";
names(Medicare)[names(Medicare)=="TOT_D"]="TOT.D";
names(Medicare)[names(Medicare)=="NUM_DCHG"]="NUM.DCHG";
names(Medicare)[names(Medicare)=="AVE_T_D"]="AVE.T.D";

Medicare$AVE.DAYS= Medicare$TOT.D/Medicare$NUM.DCHG
Medicare$CCPD=Medicare$COV.CHG/Medicare$NUM.DCHG
Medicare$NUM.DCHG=Medicare$NUM.DCHG/1000
str (Medicare)
```

```
'data.frame':   324 obs. of  11 variables:
 $ STATE   : int  1 1 1 1 1 1 2 2 2 2 ...
 $ YEAR    : int  1 2 3 4 5 6 1 2 3 4 ...
 $ TOT.CHG : num  2.21e+09 2.52e+09 2.98e+09 3.19e+09 3.42e+09 ...
 $ COV.CHG : num  2.17e+09 2.47e+09 2.92e+09 3.15e+09 3.38e+09 ...
```

```
 $ MED.REIB: num  9.73e+08 1.05e+09 1.21e+09 1.31e+09 1.38e+09 ...
 $ TOT.D   : int  1932673 1936939 2016354 1948427 1926335 1847216 51923 53051 55191 53329 ...
 $ NUM.DCHG: num  230 235 245 244 258 ...
 $ AVE.T.D : int  8 8 8 8 7 7 8 8 7 7 ...
 $ NMSTATE : Factor w/ 54 levels "AK","AL","AR",..: 2 2 2 2 2 2 1 1 1 1 ...
 $ AVE.DAYS: num  8.4 8.25 8.23 7.99 7.46 ...
 $ CCPD    : num  9435 10515 11928 12912 13098 ...
```

Some summary statistics of CCPD, NUM.DCHG, AVE>DAYS, YEAR in each year.

```
library(nlme)
attach(Medicare)
#  SUMMARY STATISTICS ;
dim(Medicare)
```

```
[1] 324  11
```

```
summary(Medicare[, c("CCPD", "NUM.DCHG", "AVE.DAYS" )])
```

```
      CCPD          NUM.DCHG          AVE.DAYS
 Min.   : 2966   Min.   :  0.515   Min.   : 5.119
 1st Qu.: 8537   1st Qu.: 42.715   1st Qu.: 7.162
 Median :10073   Median :144.282   Median : 8.067
 Mean   :10483   Mean   :210.731   Mean   : 8.542
 3rd Qu.:12059   3rd Qu.:282.884   3rd Qu.: 8.988
 Max.   :21500   Max.   :908.593   Max.   :60.251
```

```
gsummary(Medicare[, c("CCPD", "NUM.DCHG", "AVE.DAYS", "YEAR")], groups = YEAR, FUN=sd)
```

```
      CCPD NUM.DCHG AVE.DAYS YEAR
1 2466.685 202.9918 2.077437    1
2 2711.568 210.3791 7.231312    2
3 3041.274 218.9225 1.858683    3
4 3259.846 219.8253 2.112467    4
5 3345.970 226.7783 1.728882    5
6 3277.985 229.4583 1.444423    6
```

```
gsummary(Medicare[, c("CCPD", "NUM.DCHG", "AVE.DAYS", "YEAR")], groups = YEAR, FUN=mean)
```

```
       CCPD NUM.DCHG AVE.DAYS YEAR
1  8503.168 197.7274 9.048565    1
2  9472.746 203.1443 9.823055    2
3 10443.285 210.8941 8.619240    3
4 11159.680 211.2479 8.522619    4
5 11522.826 218.8690 7.898816    5
6 11796.768 222.5059 7.342360    6
```

```
gsummary(Medicare[, c("CCPD", "NUM.DCHG", "AVE.DAYS", "YEAR")], groups = YEAR, FUN=median)
```

```
       CCPD NUM.DCHG AVE.DAYS YEAR
```

```
1  7991.927 142.5880 8.533565    1
2  9113.473 142.6935 8.570416    2
3 10055.416 143.2515 8.363435    3
4 10666.865 143.6720 8.112863    4
5 10955.142 150.0765 7.560945    5
6 11171.080 152.6960 7.143355    6
```

```
gsummary(Medicare[, c("CCPD", "NUM.DCHG", "AVE.DAYS", "YEAR")], groups = YEAR, FUN=min)
```

```
      CCPD NUM.DCHG AVE.DAYS YEAR
1 3228.989    0.528 6.326762    1
2 2966.117    0.515 6.143628    2
3 3324.113    0.653 5.830248    3
4 4137.776    0.969 5.830995    4
5 4354.526    1.156 5.378061    5
6 5058.371    1.059 5.118937    6
```

```
gsummary(Medicare[, c("CCPD", "NUM.DCHG", "AVE.DAYS", "YEAR")], groups = YEAR, FUN=max)
```

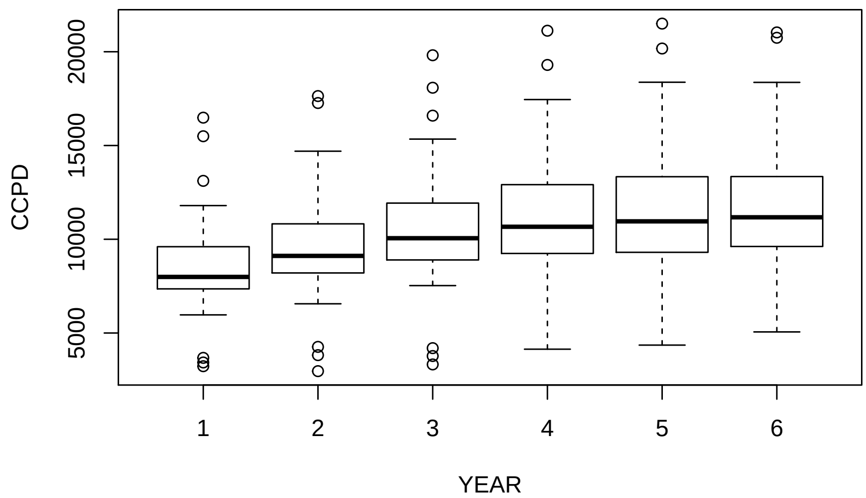
```
      CCPD NUM.DCHG AVE.DAYS YEAR
1 16484.77  849.372 17.47888    1
2 17636.51  885.919 60.25108    2
3 19814.09  908.593 16.35045    3
4 21121.55  894.216 17.13484    4
5 21500.29  905.615 14.38731    5
6 21031.58  902.479 12.79622    6
```

See the box plots of different variables in each year.

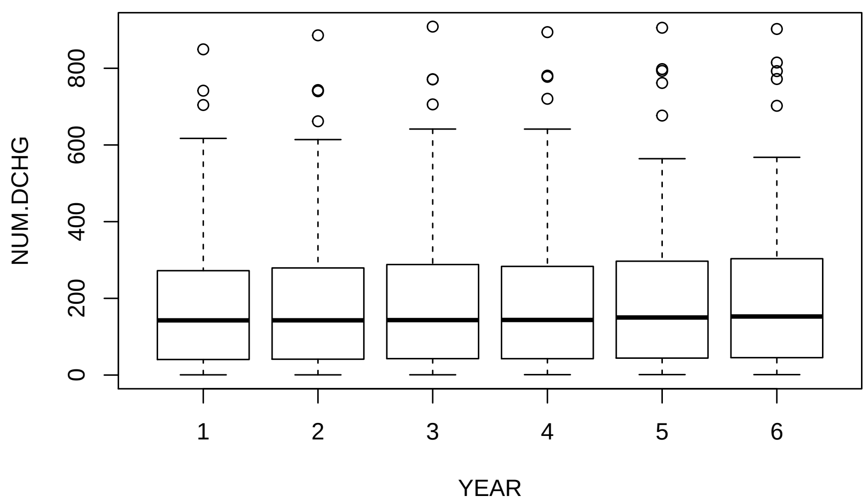
```
#  ATTACH THE DATA SET FOR SOME PRELIMINARLY LOOKS;
attach (Medicare)
```

```
The following objects are masked from Medicare (pos = 3):

    AVE.DAYS, AVE.T.D, CCPD, COV.CHG, MED.REIB, NMSTATE, NUM.DCHG,
    STATE, TOT.CHG, TOT.D, YEAR
```

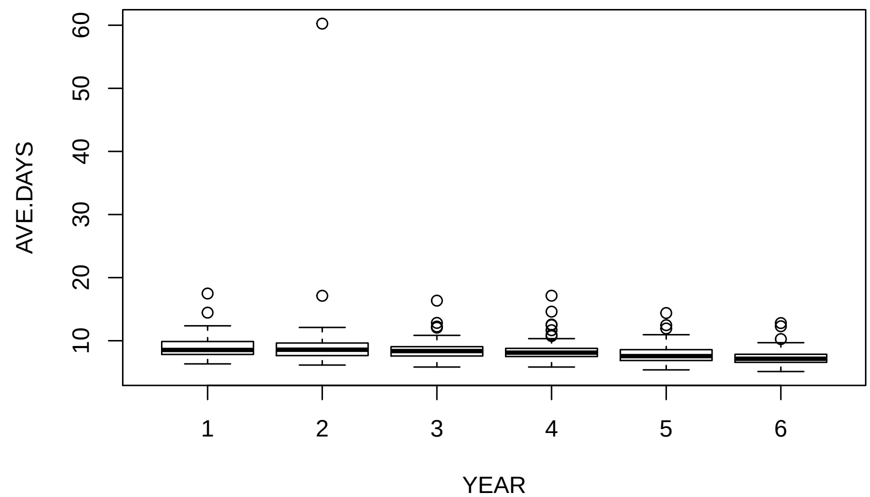
```
Medicare$YEAR=Medicare$YEAR+1989
boxplot (CCPD ~ YEAR)
```




```
boxplot (NUM.DCHG ~ YEAR)
```




```
boxplot (AVE.DAYS ~ YEAR)
```

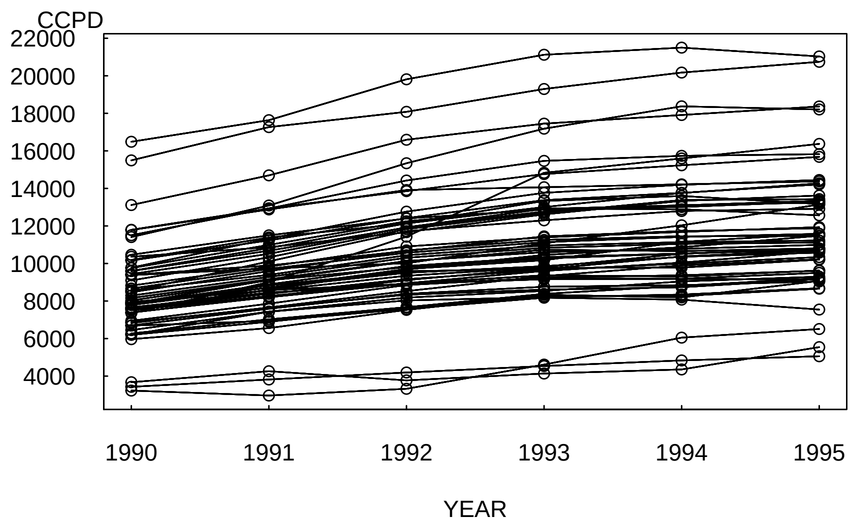


2.2 Example 2.2: Medicare Hospital Costs (Page 26)

2.2.1 FIGURE 2.1: CCPD vs YEAR; multiple time series plot

Figure 2.1 illustrates the multiple time-series plot. Here, we see that not only are overall claims increasing but also that claims increase for each state.

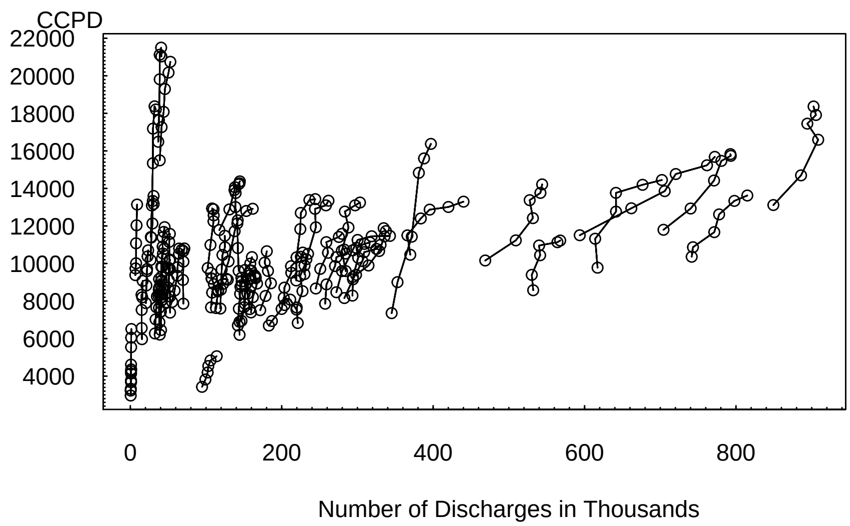
```
plot(CCPD ~ YEAR, data = Medicare, xaxt="n", yaxt="n", ylab="", xlab="")
 for (i in Medicare$STATE) {
 lines(CCPD ~ YEAR, data = subset(Medicare, STATE == i)) }
axis(2, at=seq(0, 22000, by=2000), las=1, font=10, cex=0.005, tck=0.01)
axis(1, at=seq(1990,1995, by=1), font=10, cex=0.005, tck=0.01)
mtext("CCPD", side=2, line=0, at=23000, font=12, cex=1, las=1)
mtext("YEAR", side=1, line=3, at=1992.5, font=12, cex=1)
```



2.2.2 FIGURE 2.2: CCPD vs NUM.DCHG

Figure 2.2 illustrates the scatter plot with symbols. This plot ofCCPD versus number ofdischarges, connecting observations over time, shows a positive overall relationship between CCPD and the number of discharges.

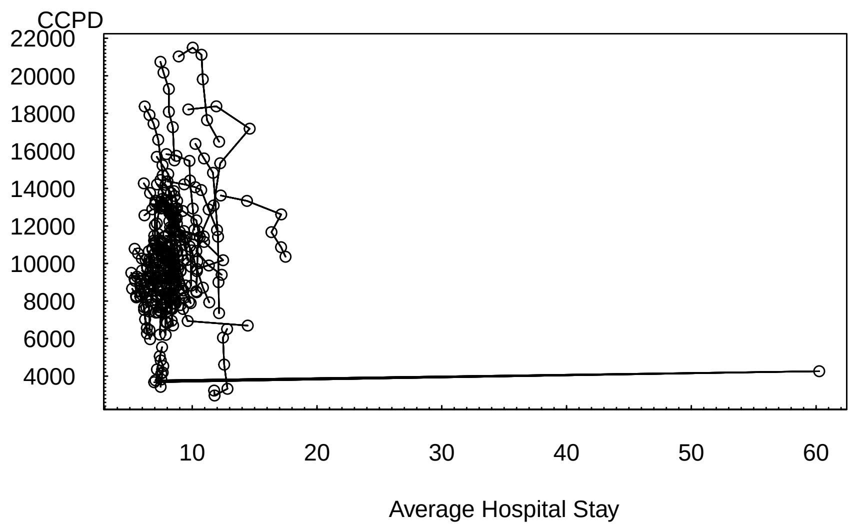
```
plot(CCPD ~ NUM.DCHG, data = Medicare, xaxt="n", yaxt="n", ylab="", xlab="")
for (i in Medicare$STATE) {
 lines(CCPD ~ NUM.DCHG, data = subset(Medicare, STATE == i)) }
axis(2, at=seq(0, 22000, by=2000), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(0, 22000, by=200), lab=F, tck=0.005)
axis(1, at=seq(0,1200, by=200), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(0,1200, by=20), lab=F, tck=0.005)
mtext("CCPD", side=2, line=0, at=23000, font=12, cex=1, las=1)
mtext("Number of Discharges in Thousands", side=1, line=3, at=500, font=12, cex=1)
```



2.2.3 Figure 2.3: CCPD vs AVE.DAYS

Figure 2.3 is a scatter plot of CCPD versus average total days, connecting observations over time. This plot demonstrates the unusual nature of the second observation for the 54th state.

```
plot(CCPD ~ AVE.DAYS, data = Medicare, ylab="", xlab="", xaxt="n", yaxt="n")
for (i in Medicare$STATE) {
 lines(CCPD ~ AVE.DAYS, data = subset(Medicare, STATE== i)) }
axis(2, at=seq(0, 22000, by=2000), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(0, 22000, by=200), lab=F, tck=0.005)
axis(1, at=seq(0,70, by=10), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(0,70, by=1), lab=F, tck=0.005)
mtext("CCPD", side=2, line=0, at=23000, font=12, cex=1, las=1)
mtext("Average Hospital Stay", side=1, line=3, at=35, font=12, cex=1)
```



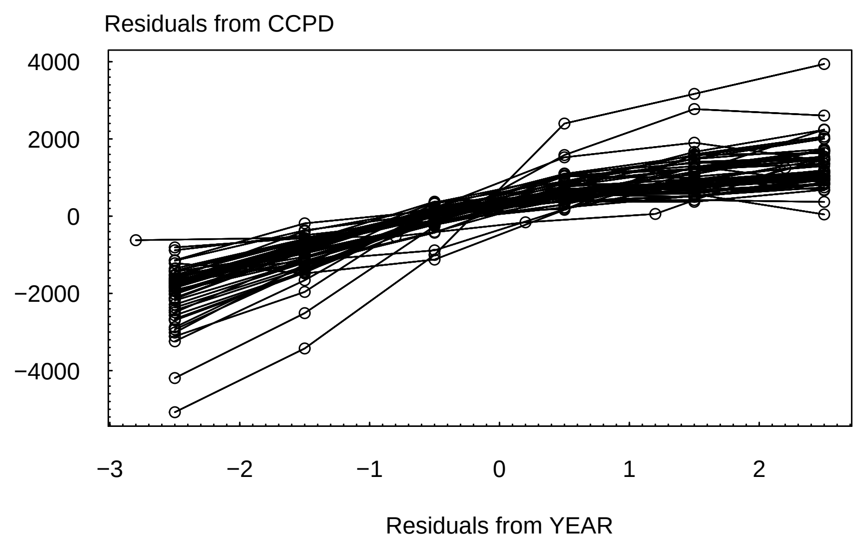
2.2.4 Figure 2.4: Added-variable plot of CCPD versus year

```
#  CREATE A CATEGORICAL VARIABLE for STATE;
Medicare$FSTATE = factor(Medicare$STATE)

#  CREATE A NEW VARIABLE;
Medicare$YEAR=Medicare$YEAR-1989
# THE NEW VARIABLES YR31 WILL BE USED IN THE FINAL MODEL TO GIVE THE 31st STATE A SPECIFIC SLOPE,
Medicare$Yr31=(Medicare$STATE==31)*Medicare$YEAR

#  CREATE A NEW DATA SET, REMOVING THE OUTLIER BY EXCLUDING THE 2ND OBSERVATION OF THE 54TH STATE
Medicare2 = subset(Medicare, STATE != 54 | YEAR != 2)
```

Figure 2.4 illustrates the basic added-variable plot. This plot portrays CCPD versus year, after excluding the second observation for the 54th state.

```
#  BASIC ADDED VARIABLE PLOT;
#  CREATE RESIDUALS;
Med1.lm = lm(CCPD ~ FSTATE, data=Medicare2)
Med2.lm = lm(YEAR ~ FSTATE, data=Medicare2)
Medicare2$rCCPD=residuals(Med1.lm)
Medicare2$rYEAR=residuals(Med2.lm)
plot(rCCPD ~ rYEAR, data=Medicare2, ylab="", xlab="", xaxt="n", yaxt="n")
for (i in Medicare2$STATE) {
```

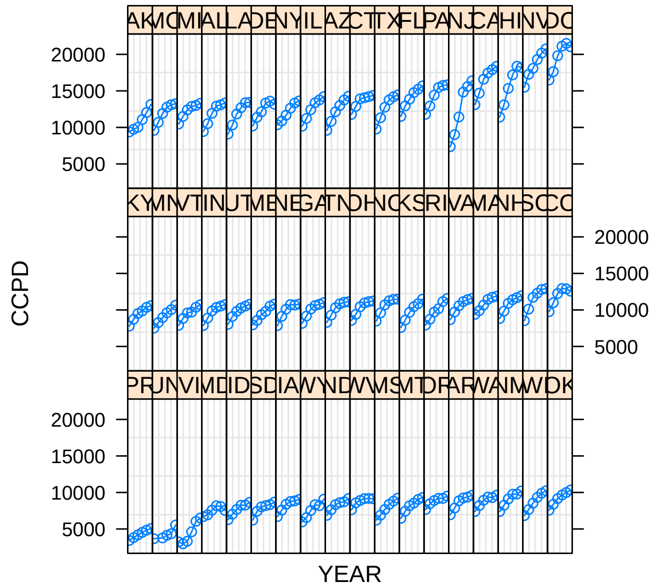
```
  lines(rCCPD ~ rYEAR, data = subset(Medicare2, STATE== i)) }
axis(2, at=seq(-6000, 4000, by=2000), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(-6000, 4000, by=200), lab=F, tck=0.005)
axis(1, at=seq(-3,3, by=1), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(-3,3, by=0.1), lab=F, tck=0.005)
mtext("Residuals from CCPD", side=2, line=-8, at=5000, font=12, cex=1, las=1)
mtext("Residuals from YEAR", side=1, line=3, at=0, font=12, cex=1)
```



2.2.5 Figure 2.5: Trellis Plot

A technique for graphical display that has recently become popular in the statistical literature is a trellis plot. This graphical technique takes its name from a trellis, which is a structure of open latticework. Figure 2.5 illustrates the use of small multiples. In each panel, the plot portrayed is identical except that it is based on a different state; this use of parallel structure allows us to demonstrate the increasing CCPD for each state.

```
GrpMedicare = groupedData(CCPD ~ YEAR| NMSTATE, data=Medicare2)
plot(GrpMedicare, xlab="YEAR", ylab="CCPD", scale = list(x=list(draw=FALSE)), layout=c
```



2.3 One way fixed effects model using lm, for linear model

See Example 2.2: Medicare Hospital Costs.

```
Medicare.lm = lm(CCPD ~ NUM.DCHG + Yr31 + YEAR + AVE.DAYS + FSTATE - 1, data=Medicare2)
summary(Medicare.lm)
```

```

Call:
lm(formula = CCPD ~ NUM.DCHG + Yr31 + YEAR + AVE.DAYS + FSTATE -
    1, data = Medicare2)

Residuals:
     Min       1Q   Median       3Q      Max
-1952.54  -264.66    50.46   300.10  1638.39

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
NUM.DCHG    10.755      2.573   4.180 3.96e-05 ***
Yr31      1262.456    128.609   9.816  < 2e-16 ***
YEAR       710.884     26.812  26.513  < 2e-16 ***
AVE.DAYS   361.290     57.979   6.231 1.81e-09 ***
FSTATE1   3888.845    894.076   4.350 1.95e-05 ***
```

```
FSTATE2   5694.017    534.048  10.662  < 2e-16 ***
FSTATE3   5736.793    661.153   8.677 4.19e-16 ***
FSTATE4   1639.697    726.577   2.257  0.02484 *
FSTATE5   1745.883   2402.770   0.727  0.46810
FSTATE6   5519.532    639.097   8.636 5.52e-16 ***
FSTATE7   5882.663    815.649   7.212 5.77e-12 ***
FSTATE8   6319.729    625.690  10.100  < 2e-16 ***
FSTATE9  12842.939    733.122  17.518  < 2e-16 ***
FSTATE10   990.386   1962.480   0.505  0.61422
FSTATE11  1352.055   1020.337   1.325  0.18628
FSTATE12  8524.447    790.980  10.777  < 2e-16 ***
FSTATE13  2700.653    475.750   5.677 3.60e-08 ***
FSTATE14  1417.162   1526.064   0.929  0.35392
FSTATE15  1383.831    952.843   1.452  0.14760
FSTATE16  1426.408    696.791   2.047  0.04163 *
FSTATE17  3146.952    673.351   4.674 4.72e-06 ***
FSTATE18  1728.006    844.017   2.047  0.04161 *
FSTATE19  3926.979    867.993   4.524 9.16e-06 ***
FSTATE20  3242.727    639.799   5.068 7.54e-07 ***
FSTATE21  -711.562    898.191  -0.792  0.42894
FSTATE22  1079.195   1161.922   0.929  0.35384
FSTATE23  2480.023   1236.826   2.005  0.04596 *
FSTATE24  2293.256    719.640   3.187  0.00161 **
FSTATE25   957.334    712.831   1.343  0.18042
FSTATE26  3299.585    998.712   3.304  0.00109 **
FSTATE27  3003.060    494.942   6.068 4.47e-09 ***
FSTATE28  3755.028    615.706   6.099 3.77e-09 ***
FSTATE29 12615.456    598.073  21.094  < 2e-16 ***
FSTATE30  4401.868    669.931   6.571 2.64e-10 ***
FSTATE31 -2649.456   1345.138  -1.970  0.04992 *
FSTATE32  3589.638    535.271   6.706 1.20e-10 ***
FSTATE33 -4444.768   2367.411  -1.877  0.06155 .
FSTATE34  1354.039   1071.140   1.264  0.20730
FSTATE35  2683.031    552.483   4.856 2.05e-06 ***
FSTATE36  -998.648   1578.677  -0.633  0.52755
FSTATE37  2109.789    736.174   2.866  0.00449 **
FSTATE38  3082.538    552.317   5.581 5.90e-08 ***
FSTATE39   260.811   2145.305   0.122  0.90333
FSTATE40 -2006.729    631.691  -3.177  0.00167 **
FSTATE41  2978.161    744.200   4.002 8.16e-05 ***
FSTATE42  3819.468    803.495   4.754 3.28e-06 ***
FSTATE43  2398.622    532.257   4.507 9.90e-06 ***
FSTATE44  1498.689   1017.547   1.473  0.14198
FSTATE45   277.739   1831.029   0.152  0.87955
FSTATE46  4580.418    496.141   9.232  < 2e-16 ***
FSTATE47  3612.284    621.289   5.814 1.75e-08 ***
```

```
FSTATE48 -2516.614    807.777  -3.115  0.00204 **
FSTATE49  2213.600    943.221   2.347  0.01967 *
FSTATE50  2138.158    685.695   3.118  0.00202 **
FSTATE51  2101.881    677.712   3.101  0.00213 **
FSTATE52  1138.089    835.624   1.362  0.17437
FSTATE53  2784.381    471.058   5.911 1.04e-08 ***
FSTATE54 -1037.318    544.265  -1.906  0.05774 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 529.5 on 265 degrees of freedom
Multiple R-squared:  0.9981,    Adjusted R-squared:  0.9977
F-statistic:  2392 on 58 and 265 DF,  p-value: < 2.2e-16
```

```
anova(Medicare.lm)
```

```
Analysis of Variance Table

Response: CCPD
           Df     Sum Sq    Mean Sq  F value    Pr(>F)
NUM.DCHG    1 2.1693e+10 2.1693e+10 77387.47 < 2.2e-16 ***
Yr31        1 6.8708e+07 6.8708e+07   245.11 < 2.2e-16 ***
YEAR        1 1.1974e+10 1.1974e+10 42716.19 < 2.2e-16 ***
AVE.DAYS    1 2.6659e+09 2.6659e+09  9510.30 < 2.2e-16 ***
FSTATE     54 2.4833e+09 4.5986e+07   164.05 < 2.2e-16 ***
Residuals 265 7.4284e+07 2.8032e+05
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.4 SECTION 2.4.1 - Analysis for the pooling test;

We can check the F-ratio by `anova(Medicare.lm,Medicare3.lm)`. We reject the null hypothesis from the result below.

```
Medicare3.lm = lm(CCPD ~ NUM.DCHG+ Yr31 + YEAR + AVE.DAYS , data=Medicare2)
summary(Medicare3.lm)
```

```
Call:
lm(formula = CCPD ~ NUM.DCHG + Yr31 + YEAR + AVE.DAYS, data = Medicare2)

Residuals:
    Min      1Q  Median      3Q     Max
-7176.7 -1255.3  -384.9  1092.4 10350.9
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 4342.1049   873.4212   4.971 1.09e-06 ***
NUM.DCHG       4.6606     0.7241   6.436 4.51e-10 ***
Yr31         299.9270   295.8341   1.014 0.311432
YEAR         733.2750    94.1398   7.789 9.62e-14 ***
AVE.DAYS     308.4710    86.0766   3.584 0.000392 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2732 on 318 degrees of freedom
Multiple R-squared:  0.2879,    Adjusted R-squared:  0.279
F-statistic: 32.15 on 4 and 318 DF,  p-value: < 2.2e-16
```

```
anova(Medicare3.lm)
```

```
Analysis of Variance Table

Response: CCPD
           Df     Sum Sq   Mean Sq F value    Pr(>F)
NUM.DCHG    1  463168764 463168764 62.0651 5.317e-14 ***
Yr31        1   33908652  33908652  4.5438 0.0338046 *
YEAR        1  366756374 366756374 49.1457 1.430e-11 ***
AVE.DAYS    1   95840842  95840842 12.8428 0.0003919 ***
Residuals 318 2373115933   7462629
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(Medicare3.lm,Medicare.lm) # pooling test
```

```
Analysis of Variance Table

Model 1: CCPD ~ NUM.DCHG + Yr31 + YEAR + AVE.DAYS
Model 2: CCPD ~ NUM.DCHG + Yr31 + YEAR + AVE.DAYS + FSTATE - 1
  Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
1    318 2373115933
2    265   74284379 53 2298831554 154.73 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.5 SECTION 2.4.2 - Correlation corresponding to the added variable plot;

As with all scatter plots, the added-variable plot can be summarized numerically through a correlation coefficient that we will denote by $corr(e_1, e_2)$.

```
#  SECTION 2.4.2 - CORRELATION CORRESPONDING TO THE ADDED VARIABLE PLOT;
library(boot)
cor(Medicare2$rCCPD , Medicare2$rYEAR)
```

```
[1] 0.8847151
```

2.6 SECTION 2.4.5 - Testing for heteroscedasticity;

When fitting regression models to data, an important assumption is that the variability is common among all observations. This assumption of common variability is called homoscedasticity, meaning "same scatter".

```
Medicare2$Resids=residuals(Medicare.lm)
Medicare2$ResidSq=Medicare2$Resids*Medicare2$Resids
MedHet.lm = lm(ResidSq ~ NUM.DCHG, data=Medicare2)
summary(MedHet.lm)
```

```

Call:
lm(formula = ResidSq ~ NUM.DCHG, data = Medicare2)

Residuals:
    Min      1Q  Median      3Q     Max 
-255383 -212155 -143167    7752 3555249 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 261171.0    35324.5   7.393 1.25e-12 ***
NUM.DCHG      -147.5      116.8  -1.264    0.207    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 454200 on 321 degrees of freedom
Multiple R-squared:  0.004949,	Adjusted R-squared:  0.001849 
F-statistic: 1.597 on 1 and 321 DF,  p-value: 0.2073
```

```
anova(MedHet.lm)
```

```
Analysis of Variance Table

Response: ResidSq
           Df     Sum Sq    Mean Sq F value Pr(>F)
NUM.DCHG    1 3.2930e+11 3.2930e+11  1.5966 0.2073
Residuals 321 6.6208e+13 2.0625e+11
```

2.6.1 One way random effects model using lm, for linear model;

We will learn random effects model in Chapter 3. Here is an example.

```
Medicare.lme = lme(CCPD ~ NUM.DCHG, data=Medicare2, random = ~1|STATE)
summary(Medicare.lme)
```

```
Linear mixed-effects model fit by REML
 Data: Medicare2
       AIC      BIC    logLik
  5733.495 5748.581 -2862.747

Random effects:
 Formula: ~1 | STATE
        (Intercept) Residual
StdDev:    3016.201 1316.346

Fixed effects: CCPD ~ NUM.DCHG
               Value Std.Error  DF   t-value p-value
(Intercept) 8084.128  564.3211 268 14.325404       0
NUM.DCHG      11.386    1.8044 268  6.310388       0
 Correlation:
         (Intr)
NUM.DCHG -0.674

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-3.6167201 -0.6276783  0.1998388  0.6325460  2.7846480

Number of Observations: 323
Number of Groups: 54
```

Chapter 3

Models with Random Effects

3.1 Import Data

```
#  "\t"  INDICATES SEPARATED BY TABLES  ;
taxprep  = read.table("TXTData/TaxPrep.txt", sep ="\t", quote = "",header=TRUE)

# taxprep=read.table(choose.files(), header=TRUE, sep="\t")
```

Data for this study are from the Statistics of Income (SOI) Panel of Individual Returns, a part of the Ernst and Young/University of Michigan Tax Research Database. The SOI Panel represents a simple random sample of unaudited individual income tax returns filed for tax years 1979-1990. The data are compiled from a stratified probability sample of unaudited individual income tax returns, Forms 1040, 1040A and 1040EZ, filed by U.S. taxpayers. The estimates that are obtained from these data are intended to represent all returns filed for the income tax years under review. All returns processed are subjected to sampling except tentative and amended returns.

Variable	Description
MS	is an indicator variable of the taxpayer's marital status. It is coded one if the taxpayer is married and zero otherwise.
HH	is an indicator variable, one if the taxpayer is a head of household and zero otherwise.
DEPEND	is the number of dependents claimed by the taxpayer.
AGE	is the presence of an indicator for age 65 or over.
F1040A	is an indicator variable of the taxpayer's filing type. It is coded one if the taxpayer uses Form 1040A and zero otherwise.

Variable	Description
F1040EZ	is an indicator variable of the taxpayer's filing type. It is coded one if the taxpayer uses Form 1040EZ and zero otherwise.
TPI	is the sum of all positive income line items on the return. is a marginal tax rate.
TXRT	is a marginal tax rate It is computed on TPI less exemptions and the standard deduction.
MR	is an exogenous marginal tax rate. It is computed on TPI less exemptions and the standard deduction.
EMP	is an indicator variable, one if Schedule C or F is present and zero otherwise. Self-employed taxpayers have greater need for professional assistance to reduce the reporting risks of doing business.
PREP	is a variable indicating the presence of a paid preparer.
TAX	is the tax liability on the return.
SUBJECT	Subject identifier, 1- 258.
TIME	Time identifier, 1-5.
LNTAX	is the natural logarithm of the tax liability on the return.
LNTPI	is the natural logarithm of the sum of all positive income line items on the return.

3.2 Example 3.2: Income Tax Payments (Page 81)

In this section, we study the effects that an individual's economic and demographic characteristics have on the amount of income tax paid. Specifically, the response of interest is `LNTAX`, defined as the natural logarithm of the liability on the tax return.

3.2.1 Table 3.2. Averages of binary variables

The binary variables in Table 3.2 indicate that over half the sample is married (MS) and approximately half the sample uses a paid preparer (PREP).

```
library(nlme)
gsummary(taxprep[, c("MS", "HH", "AGE", "EMP", "PREP")], groups=taxprep$TIME, FUN=mean)
```

```
         MS         HH        AGE       EMP      PREP
1 0.5968992 0.08139535 0.08527132 0.1395349 0.4496124
2 0.5968992 0.09302326 0.10465116 0.1589147 0.4418605
3 0.6240310 0.08527132 0.11240310 0.1550388 0.4844961
4 0.6472868 0.08139535 0.13178295 0.1472868 0.5077519
5 0.6472868 0.09302326 0.14728682 0.1472868 0.5155039
```

3.2.2 TABLE 3.3 - Summary statistics for continuous variables

Tables 3.2 and 3.3 describe the basic taxpayer characteristics used in our analysis. The summary statistics for the other nonbinary variables are in Table 3.3.

```
summary(taxprep[, c("DEPEND", "LNTPI", "MR", "LNTAX")]) #summary does not provid standard deviat
```

```
     DEPEND          LNTPI               MR            LNTAX       
 Min.   :0.000   Min.   :-0.1275   Min.   : 0.00   Min.   : 0.000  
 1st Qu.:1.000   1st Qu.: 9.4467   1st Qu.:15.00   1st Qu.: 6.645  
 Median :2.000   Median :10.0506   Median :22.00   Median : 7.701  
 Mean   :2.419   Mean   : 9.8886   Mean   :23.52   Mean   : 6.880  
 3rd Qu.:3.000   3rd Qu.:10.5320   3rd Qu.:33.00   3rd Qu.: 8.420  
 Max.   :6.000   Max.   :13.2220   Max.   :50.00   Max.   :11.860  
```

Standard deviation of some variables.

```
#Standard Deviation
var<-var(taxprep[, c("DEPEND", "LNTPI", "MR", "LNTAX")])
sqrt(diag(var))
```

```
   DEPEND     LNTPI        MR     LNTAX 
 1.337562  1.164625 11.453800  2.694961 
```

3.2.3 TABLE 3.4 - Averages by level of binary explanatory variable

To explore the relationship between each indicator variable and logarithmic tax, Table 3.4 presents the average logarithmic tax liability by level of indicator variable. This table shows that married filers pay greater tax, head-of-household filers pay less tax, taxpayers 65 or over pay less, taxpayers with self-employed income pay less, and taxpayers who use a professional tax preparer pay more.

```
library(Hmisc)
summarize(taxprep$LNTAX, taxprep$MS, mean)
```

```
  taxprep$MS taxprep$LNTAX
1          0      5.973412
2          1      7.429948
```

```
summarize(taxprep$LNTAX, taxprep$HH, mean)
```

```
  taxprep$HH taxprep$LNTAX
1          0      7.013197
2          1      5.479947
```

```
summarize(taxprep$LNTAX, taxprep$AGE, mean)
```

```
  taxprep$AGE taxprep$LNTAX
```

```
1           0      6.939184
2           1      6.430867
```

```r
summarize(taxprep$LNTAX, taxprep$EMP, mean)
```

```
  taxprep$EMP taxprep$LNTAX
1           0      6.982682
2           1      6.296879
```

```r
summarize(taxprep$LNTAX, taxprep$PREP, mean)
```

```
  taxprep$PREP taxprep$LNTAX
1            0      6.623648
2            1      7.158049
```

```r
# TABLE counts of BINARY EXPLANATORY VARIABLE
# CREATE CATEGORICAL VARIABLE
taxprep$MSF=taxprep$MS
taxprep$HHF=taxprep$HH
taxprep$AGEF=taxprep$AGE
taxprep$EMPF=taxprep$EMP
taxprep$PREPF=taxprep$PREP
table(taxprep$MSF)
```

```

  0   1
487 803
```

```r
table(taxprep$HHF)
```

```

   0    1
1178  112
```

```r
table(taxprep$AGEF)
```

```

   0    1
1140  150
```

```r
table(taxprep$EMPF)
```

```

   0    1
1097  193
```

```r
table(taxprep$PREPF)
```

```

  0   1
671 619
```

3.2.4 TABLE 3.5 - Correlation for continous variables

Table 3.5 summarizes basic relations among logarithmic tax and the other non-binary explanatory variables. Both `LNTPI` and `MR` are strongly correlated with logarithmic tax whereas the relationship between `DEPEND` and logarithmic tax is positive, yet weaker. Table 3.5 also shows that `LNTPI` and `MR` are strongly positively correlated.

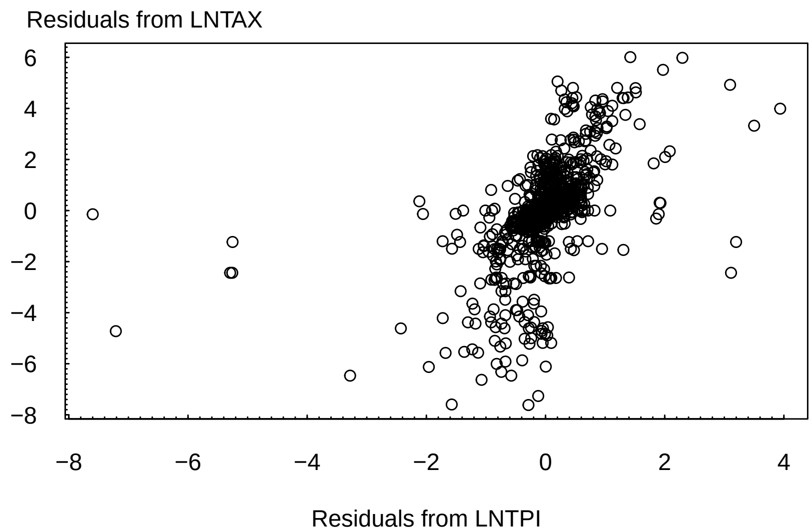
```
cor(taxprep[,c("LNTAX", "DEPEND", "LNTPI", "MR")])
```

```
            LNTAX     DEPEND     LNTPI        MR
LNTAX  1.00000000 0.08519899 0.7176476 0.7466574
DEPEND 0.08519899 1.00000000 0.2777381 0.1275044
LNTPI  0.71764760 0.27773808 1.0000000 0.7958007
MR     0.74665744 0.12750438 0.7958007 1.0000000
```

3.2.5 FIGURE 3.2: Basic added variable plot (y vs. x)

Moreover, both the mean and median marginal tax rates (`MR`) are decreasing, although mean and median tax liabilities (`LNTAX`) are stable (see Figure 3.2). These results are consistent with congressional efforts to reduce rates and expand the tax base through broadening the definition of income and eliminating deductions.

```
#CREATE CATEGORICAL VARIABLE
taxprep$SUBJECT1=factor(taxprep$SUBJECT)
lntax.lm = lm(LNTAX ~ SUBJECT1, data=taxprep)
lntpi.lm = lm(LNTPI ~ SUBJECT1, data=taxprep)
taxprep$Resid1=residuals(lntax.lm)
taxprep$Resid2=residuals(lntpi.lm)
plot(Resid1 ~ Resid2, data=taxprep, xaxt="n", yaxt="n", ylab="", xlab="")
axis(2, at=seq(-8, 7, by=2), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(-8, 8, by=0.2), lab=F, tck=0.005)
axis(1, at=seq(-8,4, by=2), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(-8, 4, by=0.2), lab=F, tck=0.005)
mtext("Residuals from LNTAX", side=2, line=-7, at=7.5, font=10, cex=1, las=1)
mtext("Residuals from LNTPI", side=1, line=3, at=-2, font=10, cex=1)
```



3.2.6 DISPLAY 3.1 - Error components model

The estimated model appears in Display 3.1, from a fit using the statistical package SAS. Display 3.1 shows that HH, EMP, LNTPI, and MR are statistically significant variables that affect LNTAX. Somewhat surprisingly, the PREP variable was not statistically significant.

```
random<-lme(LNTAX~MS+HH+AGE+EMP+PREP+LNTPI+DEPEND+MR, data=taxprep, random=~1|SUBJECT,
## NOTE* THE DEFAULT METHOD IN lme IS "REML"
summary(random)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: taxprep
       AIC      BIC    logLik
  4813.255 4870.041 -2395.627

Random effects:
 Formula: ~1 | SUBJECT
        (Intercept) Residual
StdDev:   0.9602161 1.368896

Fixed effects: LNTAX ~ MS + HH + AGE + EMP + PREP + LNTPI + DEPEND + MR
                 Value Std.Error   DF   t-value p-value
(Intercept) -2.9603371 0.5705536 1024 -5.188534  0.0000
MS           0.0373000 0.1824839 1024  0.204402  0.8381
```

```
HH          -0.6889876 0.2320057 1024 -2.969702  0.0031
AGE          0.0207431 0.2000035 1024  0.103713  0.9174
EMP         -0.5048035 0.1679848 1024 -3.005054  0.0027
PREP        -0.0217036 0.1175229 1024 -0.184675  0.8535
LNTPI        0.7604058 0.0699692 1024 10.867728  0.0000
DEPEND      -0.1127475 0.0592818 1024 -1.901891  0.0575
MR           0.1153752 0.0073142 1024 15.774213  0.0000
 Correlation:
       (Intr) MS     HH     AGE    EMP    PREP   LNTPI  DEPEND
MS      0.176
HH      0.030  0.419
AGE    -0.043 -0.167 -0.023
EMP    -0.116 -0.069  0.024 -0.030
PREP   -0.035 -0.045  0.004 -0.115 -0.112
LNTPI  -0.948 -0.180 -0.081 -0.043  0.099 -0.016
DEPEND -0.074 -0.604 -0.269  0.224 -0.038 -0.039 -0.068
MR      0.522 -0.020  0.055  0.149 -0.041 -0.051 -0.698  0.102

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-5.83483692 -0.21263981  0.09677632  0.39814646  5.79731648

Number of Observations: 1290
Number of Groups: 258
```

3.3 SECTION 3.3 - Random coefficients model

```
#randomcoeff<-lme(LNTAX~MS+HH+AGE+EMP+PREP+LNTPI+DEPEND+MR, data=taxprep, random=~1+MS+HH+AGE+EMP
# NOTE*:It takes forever to run the estimation, in the end a warning messaged was given.
# No estimation result was produced.
# The reason is due to the fact that in SAS, the method of mivque0 allows estimation for this mod
```

Chapter 4

Prediction and Bayesian Inference

4.1 Import Data

```
lottery  = read.table("TXTData/Lottery.txt", sep ="\t", quote = "",header=TRUE)

#lottery=read.table(choose.files(), header=TRUE, sep="\t")
```

State of Wisconsin lottery administrators provided weekly lottery sales data. We consider online lottery tickets that are sold by selected retail establishments in Wisconsin. These tickets are generally priced at $1.00, so the number of tickets sold equals the lottery revenue. We analyze lottery sales (`OLSALES`) over a forty-week period, April, 1998 through January, 1999, from fifty randomly selected ZIP codes within the state of Wisconsin. We also consider the number of retailers within a ZIP code for each time (`NRETAIL`).

Variable	Description
OLSALES	Online lottery sales to individual consumers
NRETAIL	Number of listed retailers
PERPERHH	Persons per household MEDSCHYR Median years of schooling
MEDHVL	Median home value in $1000s for owner-occupied homes PRCRENT
PRC55P	Percent of population that is 55 or older
HHMEDAGE	Household median age
MEDINC	Estimated median household income, in $1000s
POPULATN	Population, in thousands

```
#EXTRACT TIME - INVARIANT INFORMATION TO ANALYZE
mzip=d=as.data.frame(t(sapply(split(lottery[, c("NRETAIL", "PERPERHH", "OLSALES", "MEDS
 # Extract time invariant information to analyze
# Notice: the code for this part on website is wrong.
```

4.2 Example: Forecasting Wisconsin Lottery Sales (Page 138)

In this section, we forecast the sale of state lottery tickets from 50 postal (ZIP) codes inWisconsin. Lottery sales are an important component of state revenues. Accurate forecasting helps in the budget-planning process. A model is useful in assessing the important determinants of lottery sales, and understanding the determinants of lottery sales is useful for improving the design of the lottery sales system. Additional details of this study are in Frees and Miller (2003O).

4.2.1 TABLE 4.2: Time - invariant summary statistics

```
summary(mzip[,c("NRETAIL", "PERPERHH", "OLSALES", "MEDSCHYR", "MEDHVL", "PRCRENT", "PRC
```

```
    NRETAIL          PERPERHH         OLSALES           MEDSCHYR
 Min.   : 1.000   Min.   :2.200   Min.   :  189.0   Min.   :12.20
 1st Qu.: 3.000   1st Qu.:2.600   1st Qu.:  821.3   1st Qu.:12.50
 Median : 6.362   Median :2.700   Median : 2426.4   Median :12.60
 Mean   :11.942   Mean   :2.706   Mean   : 6494.8   Mean   :12.70
 3rd Qu.:15.312   3rd Qu.:2.800   3rd Qu.:10016.5   3rd Qu.:12.78
 Max.   :68.625   Max.   :3.200   Max.   :33181.4   Max.   :15.90
     MEDHVL          PRCRENT          PRC55P        HHMEDAGE
 Min.   : 34.50   Min.   : 6.00   Min.   :25.0   Min.   :41.00
 1st Qu.: 43.77   1st Qu.:19.25   1st Qu.:35.0   1st Qu.:46.00
 Median : 53.90   Median :24.00   Median :40.0   Median :48.00
 Mean   : 57.09   Mean   :24.68   Mean   :39.7   Mean   :48.76
 3rd Qu.: 66.47   3rd Qu.:27.00   3rd Qu.:44.0   3rd Qu.:51.00
 Max.   :120.00   Max.   :62.00   Max.   :56.0   Max.   :59.00
     MEDINC         POPULATN
 Min.   :27.90   Min.   : 0.280
 1st Qu.:38.17   1st Qu.: 1.964
 Median :43.10   Median : 4.405
 Mean   :45.12   Mean   : 9.311
 3rd Qu.:53.62   3rd Qu.:15.446
 Max.   :70.70   Max.   :39.098
```

```
# STANDARD DEVIATION
sqrt(diag(var(mzip[,c("NRETAIL", "PERPERHH", "OLSALES", "MEDSCHYR", "MEDHVL", "PRCRENT"
```

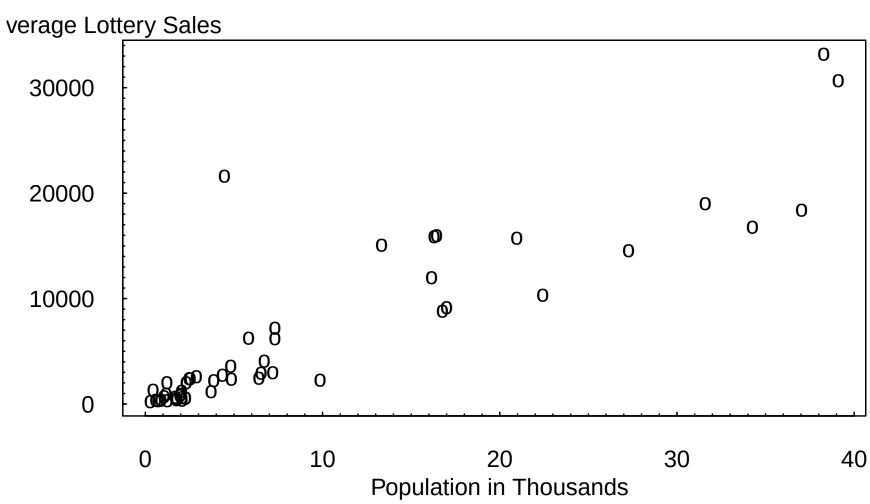
```
     NRETAIL     PERPERHH      OLSALES     MEDSCHYR        MEDHVL
```

```
  13.2918231     0.2093820 8103.0125037     0.5514212   18.3731152
     PRCRENT        PRC55P     HHMEDAGE        MEDINC     POPULATN
   9.3425513     7.5112161    4.1431527     9.7835616   11.0981570
```

4.2.2 FIGURE 4.2: Look at the relationship

Figure 4.2 shows a positive relationship between average online sales and population. Further, the ZIP code corresponding to the city of Kenosha, Wisconsin, has unusually large average sales for its population size.

```
plot(OLSALES ~ POPULATN, data = mzip, xlab="", ylab="", xaxt="n", yaxt="n",pch="o", las=1, cex=1)

axis(2, at=seq(0, 40000, by=10000), las=1, font=10, cex=0.005, tck=0.01)

axis(2, at=seq(0, 40000, by=1000), lab=F, tck=0.005)

axis(1, at=seq(0,40, by=10), font=10, cex=0.005, tck=0.01)

axis(1, at=seq(0,40, by=1), lab=F, tck=0.005)

mtext("Average Lottery Sales", side=2, line=-3.5, at=36000, font=10, cex=1, las=1)

mtext("Population in Thousands", side=1, line=2, at=20, font=10, cex=1, las=1)
```

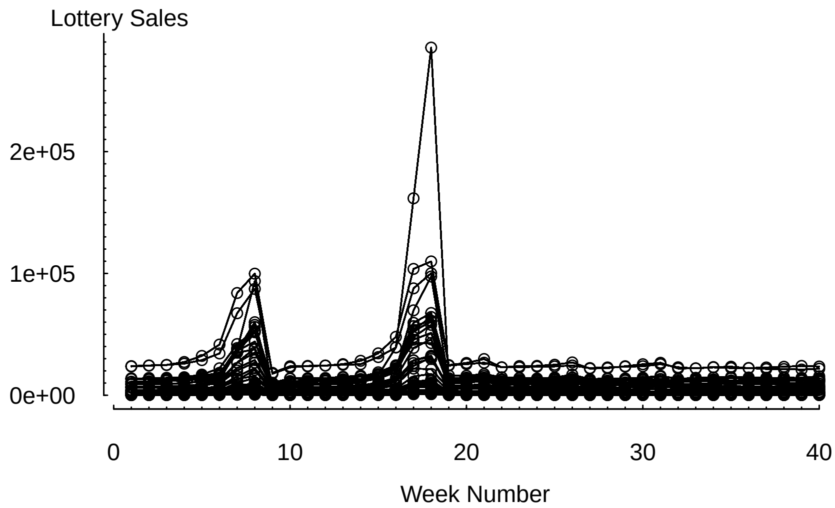


```
lottery$logsales<-log10(lottery$OLSALES)
m<-order(lottery$ZIP, lottery$TIME, lottery$OLSALES,lottery$logsales)

index<-as.data.frame(cbind(lottery$ZIP[m],lottery$TIME[m],lottery$OLSALES[m],lottery$lo

names(index)<-c("ZIP", "TIME", "OLSALES", "LOGSALES")
```

4.2.4 FIGURE 4.3: Lottery vs. week number

Figure 4.3 presents a multiple time-series plot of (weekly) sales over time. Here, each line traces the sales patterns for a particular ZIP code. This figure shows the dramatic increase in sales for most ZIP codes, at approximately weeks 8 and 18.

```
plot(OLSALES ~ TIME, data = lottery, axes=F, ylab="", xlab="", xaxt="n", yaxt="n")
for (i in index$ZIP) {
     lines(OLSALES ~ TIME, data = subset(index, ZIP == i)) }
axis(1, at=seq(0,40, by=1), labels=F, tck=0.005)
axis(1, at=seq(0,40, by=10), cex=0.005, tck=0.01)
mtext("Week Number", side=1, line=2.5, cex=1, font=10)
axis(2, at=seq(0, 300000, by=10000), labels=F, tck=0.005)
axis(2, at=seq(0, 305000, by=100000), las=1, cex=0.005, tck=0.01)
mtext("Lottery Sales", side=2, line=-3, at=310000, font=10, cex=1, las=1)
```



Another way of producing multiple time series graph by using trellis xyplot:

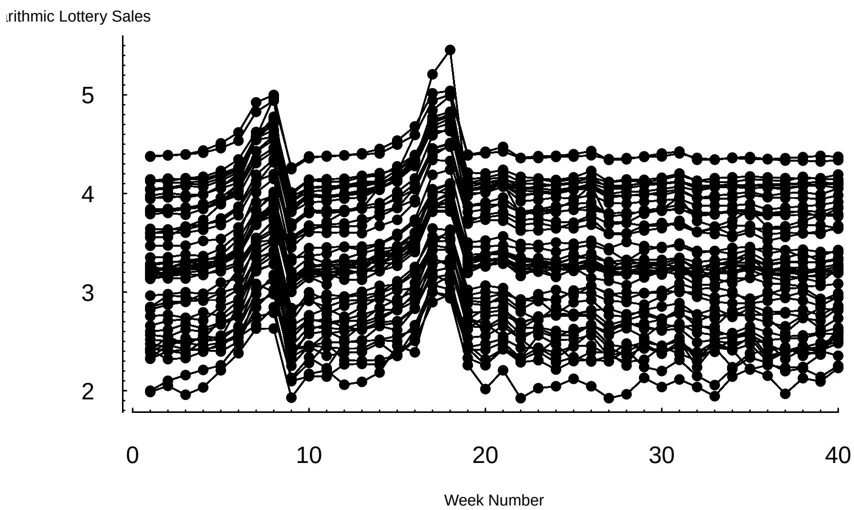
```
library(lattice)
trellis.device(color=F) # telling the trellis device to mimic 'black and white'
xyplot(OLSALES ~ TIME, data=index, groups=ZIP, scales=list(y=list(at=seq(0, 300000,100000), tck=.

#ChECK LOG VALUES
lottery$logsales<-log10(lottery$OLSALES)
lottery$lnsales<-log(lottery$OLSALES)
```

4.2.5 FIGURE 4.4: Log lottery vs week number

Figure 4.4 shows the same information as in Figure 4.3 but on a common (base 10) logarithmic scale. Here, we still see the effects of the PowerBall jackpots on sales. However, Figure 4.4 suggests a dynamic pattern that is common to all ZIP codes. Specifically, logarithmic sales for each ZIP code are relatively stable with the same approximate level of variability. Further, logarithmic sales for each ZIP code peak at the same time, corresponding to large PowerBall jackpots.

```
#FIGURE 4.4 LOG LOTTERY vs WEEK NUMBER
plot(LOGSALES ~ TIME, data = index, type="p", axes=F, ylab="", xlab="", pch=16, mkh=0.0001, lwd=0
axis(1, at=seq(0,40, by=1), labels=F, tck=0.005)
axis(1, at=seq(0,40, by=10), cex=0.4, tck=0.01)
mtext("Week Number", side=1, line=2.5, cex=0.7, font=10)
axis(2, at=seq(0, 6, by=0.1), labels=F, tck=0.005)
```

```
axis(2, at=seq(0, 6, by=1), las=1, cex=0.4, tck=0.01)
mtext("Logarithmic Lottery Sales", side=2, line=-1, at=5.8, font=10, cex=0.7, las=1)
    for (i in index$ZIP) {
    lines(LOGSALES ~ TIME, data=subset(index, ZIP==i)) }
```



4.3 Create model development sample

```
Lottery=lottery
Lottery$LNSALES<-log(Lottery$OLSALES)
Lottery2<-subset(Lottery, Lottery$TIME<36)
```

4.3.1 MODEL 1. Pooled cross-setional model

```
lm1<-lm(LNSALES~PERPERHH+MEDSCHYR+MEDHVL+PRCRENT+PRC55P+HHMEDAGE+MEDINC+POPULATN+NRETAI
summary(lm1)
```

```
Call:
lm(formula = LNSALES ~ PERPERHH + MEDSCHYR + MEDHVL + PRCRENT +
    PRC55P + HHMEDAGE + MEDINC + POPULATN + NRETAIL, data = Lottery2)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-1.9743 -0.6012 -0.0774  0.5430  4.2015

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.821060   1.339594  10.317  < 2e-16 ***
PERPERHH    -1.084705   0.160224  -6.770 1.76e-11 ***
MEDSCHYR    -0.821644   0.069049 -11.899  < 2e-16 ***
MEDHVL       0.013822   0.002662   5.192 2.33e-07 ***
PRCRENT      0.031820   0.003738   8.512  < 2e-16 ***
PRC55P      -0.069578   0.013397  -5.194 2.30e-07 ***
HHMEDAGE     0.118136   0.020961   5.636 2.03e-08 ***
MEDINC       0.043373   0.005304   8.177 5.53e-16 ***
POPULATN     0.057025   0.006060   9.410  < 2e-16 ***
NRETAIL      0.021278   0.004076   5.220 2.00e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8365 on 1740 degrees of freedom
Multiple R-squared:  0.6963,    Adjusted R-squared:  0.6947
F-statistic: 443.3 on 9 and 1740 DF,  p-value: < 2.2e-16
```

4.3.2 MODEL 2. Error components model

```r
library(nlme)
lme1<-lme(LNSALES~PERPERHH+MEDSCHYR+MEDHVL+PRCRENT+PRC55P+HHMEDAGE+MEDINC+POPULATN+NRETAIL, data=
# NOTE* THE DEFAULT METHOD IN lme IS "REML"
# Use REML method in estimating fixed effects beta coefficients
summary(lme1)
```

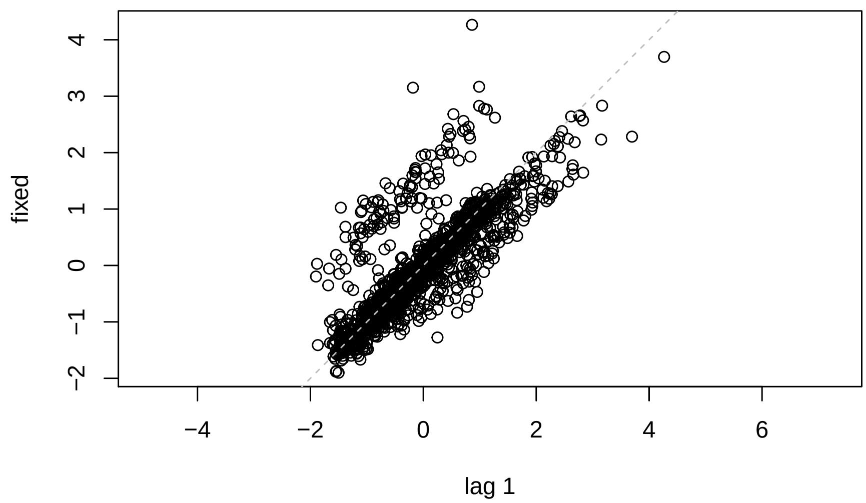
```
Linear mixed-effects model fit by REML
 Data: Lottery2
       AIC      BIC    logLik
  2907.889 2973.428 -1441.944

Random effects:
 Formula: ~1 | ZIP
        (Intercept)  Residual
StdDev:     0.77897 0.5130729

Fixed effects: LNSALES ~ PERPERHH + MEDSCHYR + MEDHVL + PRCRENT + PRC55P + HHMEDAGE +      MEDINC
                Value Std.Error   DF   t-value p-value
(Intercept) 18.095695  7.316764 1699  2.473183  0.0135
PERPERHH    -1.287021  0.886172   41 -1.452337  0.1540
MEDSCHYR    -1.077937  0.375131   41 -2.873491  0.0064
MEDHVL       0.007360  0.014633   41  0.502935  0.6177
```

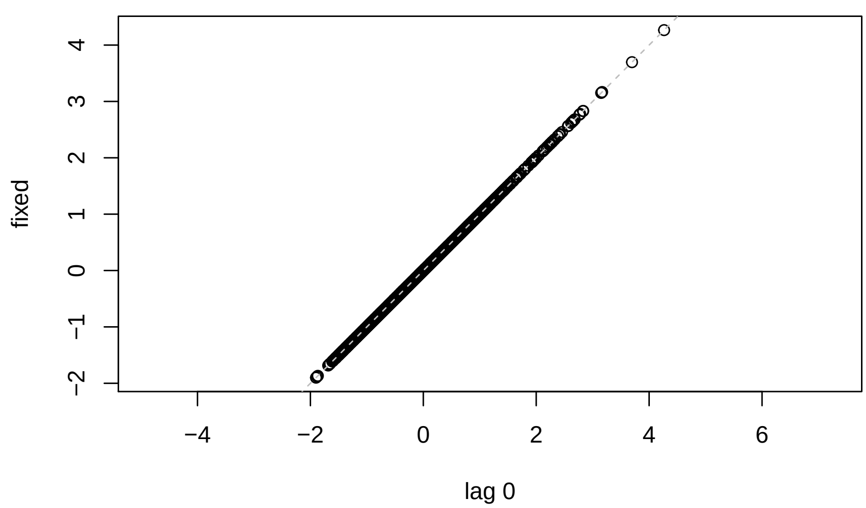
```
PRCRENT      0.026321  0.020660   41  1.274032  0.2098
PRC55P      -0.072547  0.074259   41 -0.976939  0.3343
HHMEDAGE     0.118637  0.116199   41  1.020986  0.3132
MEDINC       0.045540  0.029396   41  1.549194  0.1290
POPULATN     0.121851  0.027529   41  4.426231  0.0001
NRETAIL     -0.027177  0.017420 1699 -1.560055  0.1189
 Correlation:
         (Intr) PERPER MEDSCH MEDHVL PRCREN PRC55P HHMEDA MEDINC POPULA
PERPERHH -0.632
MEDSCHYR -0.745  0.204
MEDHVL    0.303  0.093 -0.394
PRCRENT  -0.198  0.402 -0.258  0.008
PRC55P    0.146  0.236 -0.018  0.069  0.039
HHMEDAGE -0.461  0.049  0.109 -0.128  0.151 -0.898
MEDINC   -0.171 -0.013  0.080 -0.653  0.214  0.392 -0.200
POPULATN  0.180 -0.021 -0.228 -0.171 -0.287 -0.035  0.035 -0.050
NRETAIL  -0.210  0.082  0.246  0.159  0.096  0.014 -0.002 -0.027 -0.847

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.06921597 -0.49881173 -0.29717361  0.02767368  6.60150830

Number of Observations: 1750
Number of Groups: 50
```

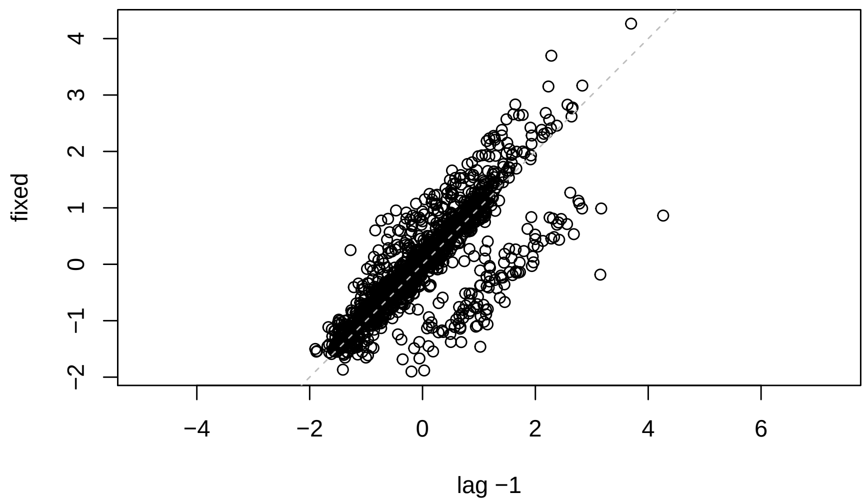
```
# CHECK AUTOCORRELATION PATTERNS
ACF(lme1, maxlag=10) #Obtain ACF of residuals from lme1
```

```
   lag         ACF
1    0  1.00000000
2    1  0.52724776
3    2  0.10000857
4    3 -0.03788895
5    4 -0.15969734
6    5 -0.23410399
7    6 -0.24984691
8    7 -0.18355756
9    8 -0.02825433
10   9  0.19456638
11  10  0.47143962
12  11  0.17601528
13  12 -0.10862546
14  13 -0.16449717
15  14 -0.31225995
16  15 -0.40819498
```

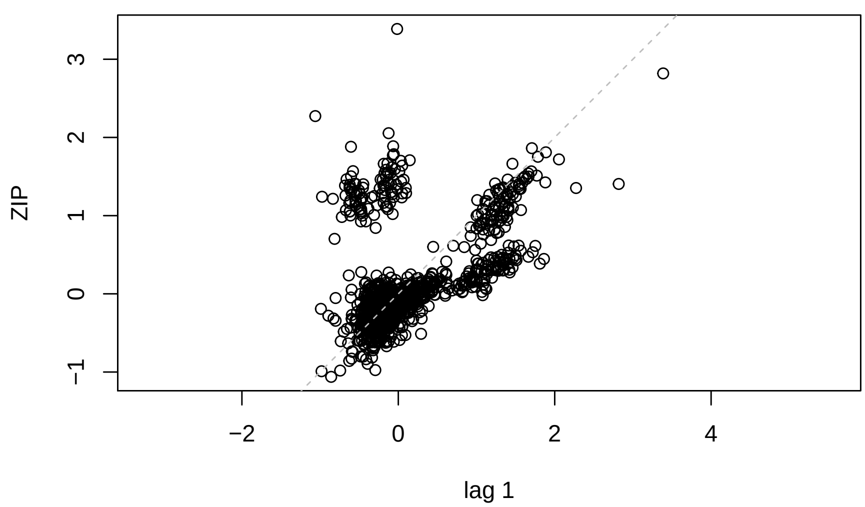
```
lag.plot(lme1$residuals, lags=-1) #Autocorrelation patterns one lag, needs to refine
```



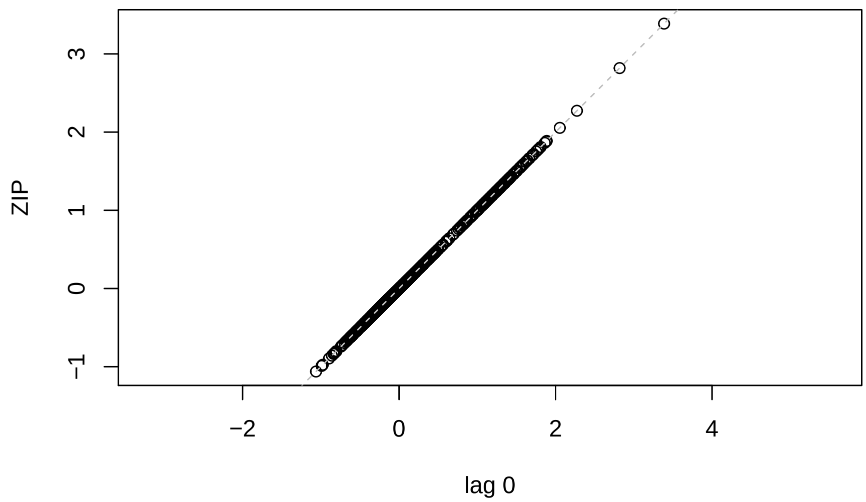


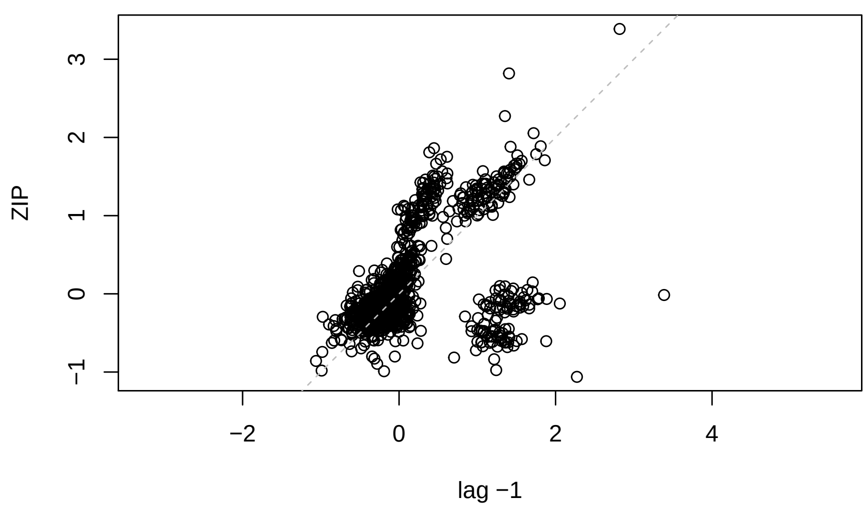


fixed
−2
−1
0
1
2
3
4
−4
−2
0
2
4
6
lag −1



ZIP
−1
0
1
2
3
−2
0
2
4
lag 1


ZIP
3
2
1
0
−1
−2
0
2
4
lag 0


ZIP
3
2
1
0
−1
−2
0
2
4
lag −1

4.3.3 MODEL 3. Error components model with autocorrelated errors

```
lme2<-update(lme1, correlation=corAR1(form=~TIME|ZIP))
summary(lme2)
```

```
Linear mixed-effects model fit by REML
 Data: Lottery2
       AIC      BIC    logLik
  2318.834 2389.835 -1146.417

Random effects:
 Formula: ~1 | ZIP
        (Intercept)  Residual
StdDev:    0.726541 0.5282642

Correlation Structure: AR(1)
 Formula: ~TIME | ZIP
 Parameter estimate(s):
      Phi
0.5552575
Fixed effects: LNSALES ~ PERPERHH + MEDSCHYR + MEDHVL + PRCRENT + PRC55P + HHMEDAGE +
                Value Std.Error   DF    t-value p-value
(Intercept) 15.254535  7.005477 1699  2.1775157  0.0296
PERPERHH    -1.149312  0.842554   41 -1.3640808  0.1800
MEDSCHYR    -0.911242  0.360225   41 -2.5296504  0.0154
MEDHVL       0.011273  0.013960   41  0.8074825  0.4240
PRCRENT      0.030104  0.019652   41  1.5319015  0.1332
PRC55P      -0.071434  0.070515   41 -1.0130333  0.3170
HHMEDAGE     0.119779  0.110336   41  1.0855851  0.2840
MEDINC       0.044082  0.027916   41  1.5790867  0.1220
POPULATN     0.080430  0.029449   41  2.7311900  0.0093
NRETAIL      0.003887  0.019402 1699  0.2003424  0.8412
 Correlation:
         (Intr) PERPER MEDSCH MEDHVL PRCREN PRC55P HHMEDA MEDINC POPULA
PERPERHH -0.632
MEDSCHYR -0.750  0.209
MEDHVL    0.286  0.097 -0.373
PRCRENT  -0.204  0.403 -0.246  0.014
PRC55P    0.144  0.236 -0.017  0.069  0.039
HHMEDAGE -0.457  0.049  0.108 -0.128  0.151 -0.898
MEDINC   -0.167 -0.014  0.077 -0.652  0.212  0.392 -0.200
POPULATN  0.217 -0.042 -0.269 -0.196 -0.281 -0.035  0.032 -0.037
NRETAIL  -0.245  0.097  0.285  0.185  0.112  0.017 -0.002 -0.031 -0.881

Standardized Within-Group Residuals:
```

```
        Min          Q1         Med          Q3         Max 
-1.87105785 -0.46121477 -0.26278521  0.04905521  6.55032232 

Number of Observations: 1750
Number of Groups: 50 
```

4.3.4 MODEL 4. More parsimonious random effects model

```
lme3<-lme(LNSALES~MEDSCHYR+POPULATN, data=Lottery2, random=~1|ZIP, correlation=corAR1(form=~TIME|
summary(lme3)
```

```
Linear mixed-effects model fit by REML
 Data: Lottery2 
       AIC      BIC    logLik
  2291.584 2324.378 -1139.792

Random effects:
 Formula: ~1 | ZIP
        (Intercept)  Residual
StdDev:    0.838855 0.5280303

Correlation Structure: AR(1)
 Formula: ~TIME | ZIP 
 Parameter estimate(s):
      Phi 
0.5549028 
Fixed effects: LNSALES ~ MEDSCHYR + POPULATN 
                Value Std.Error   DF   t-value p-value
(Intercept)  7.983814  3.407381 1700  2.343094  0.0192
MEDSCHYR    -0.097917  0.273978   47 -0.357391  0.7224
POPULATN     0.108468  0.013613   47  7.968097  0.0000
 Correlation: 
         (Intr) MEDSCH
MEDSCHYR -0.999       
POPULATN  0.565 -0.590

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max 
-1.81585940 -0.45547600 -0.25704219  0.06499433  6.60925098 

Number of Observations: 1750
Number of Groups: 50 
```

```
#THE POOLED CROSS-SECTIONAL MODEL WITH AUTOCORRELATED ERRORS
#Default method for gls is reml, gls can be viewed as an lme function without the argument random
```

```
gls1<-gls(LNSALES~PERPERHH+MEDSCHYR+MEDHVL+PRCRENT+PRC55P+HHMEDAGE+MEDINC+POPULATN+NRE
gls1
```

```
Generalized least squares fit by REML
  Model: LNSALES ~ PERPERHH + MEDSCHYR + MEDHVL + PRCRENT + PRC55P + HHMEDAGE +      MI
  Data: Lottery2
  Log-restricted-likelihood: -1240.823

Coefficients:
(Intercept)    PERPERHH    MEDSCHYR      MEDHVL     PRCRENT      PRC55P
13.59613280 -1.06322470 -0.82003019  0.01293351  0.03252353 -0.07218282
   HHMEDAGE      MEDINC    POPULATN     NRETAIL
 0.12285161  0.04322324  0.05898877  0.02014254

Correlation Structure: AR(1)
 Formula: ~TIME | ZIP
 Parameter estimate(s):
      Phi
0.8240088
Degrees of freedom: 1750 total; 1740 residual
Residual standard error: 0.8427768
```

4.3.5 MODEL 5. Fixed effects model with autocorrelated errors

```
Lottery2$ZIPfac=factor(Lottery2$ZIP)
gls2<-gls(LNSALES~ZIPfac, data=Lottery2, correlation=corAR1(form=~TIME|ZIPfac))
gls2
```

```
Generalized least squares fit by REML
  Model: LNSALES ~ ZIPfac
  Data: Lottery2
  Log-restricted-likelihood: -1073.063

Coefficients:
(Intercept) ZIPfac53033 ZIPfac53038 ZIPfac53059 ZIPfac53072 ZIPfac53083
 7.02034302  1.07644720  0.61877604 -0.01329196  2.45963633  1.97648544
ZIPfac53095 ZIPfac53098 ZIPfac53104 ZIPfac53172 ZIPfac53211 ZIPfac53520
 3.27341024  0.59051473  2.12004892  2.56764964  2.68989060  1.53954270
ZIPfac53544 ZIPfac53563 ZIPfac53572 ZIPfac53574 ZIPfac53813 ZIPfac53924
-1.22340317  1.68388653  0.81336415  0.64605315  0.79960910 -1.46385511
ZIPfac53934 ZIPfac53943 ZIPfac53952 ZIPfac54115 ZIPfac54143 ZIPfac54153
 0.65471544 -1.31041280  0.49735851  2.03324306  2.40463509  1.12075652
ZIPfac54170 ZIPfac54205 ZIPfac54213 ZIPfac54220 ZIPfac54235 ZIPfac54241
-0.09585752 -0.80190426 -0.69482488  3.22278219  2.21634671  1.96067334
```

```
ZIPfac54302 ZIPfac54406 ZIPfac54436 ZIPfac54457 ZIPfac54470 ZIPfac54474
 2.39541360  0.75120386 -1.31334267  1.60733345 -0.28003078  0.51088577
ZIPfac54480 ZIPfac54531 ZIPfac54556 ZIPfac54614 ZIPfac54622 ZIPfac54634
-1.62581025 -0.73327749 -0.37613230 -0.50434660 -0.79801244  0.49941189
ZIPfac54650 ZIPfac54701 ZIPfac54724 ZIPfac54745 ZIPfac54758 ZIPfac54810
 2.50724987  2.72121016  0.66159018 -0.99678783  0.71914194 -1.16821041
ZIPfac54839 ZIPfac54956
-2.00207902  2.57602144

Correlation Structure: AR(1)
 Formula: ~TIME | ZIPfac
 Parameter estimate(s):
    Phi
0.55476
Degrees of freedom: 1750 total; 1700 residual
Residual standard error: 0.5279669
```

```
# Note the difference between R estimates and SAS estimates is because in SAS the estimate
# for ZIP 54956 is restricted to be zero, in R the intercept and estimates for Zip are
# scaled differently, but both estimates should give us approximately the same answer#
```

The five models listed are summarized in Table 4.4 at Page 146.

Chapter 5

Multilevel Models

5.1 Import Data

```
#Dental=read.table(choose.files(), header=TRUE, sep="\t")
#library(mice)
#data(potthoffroy)
# I make this dataset myself according to data(potthoffroy)
Dental <- read.table("TXTData/dental.txt",sep ="\t", quote = "", header=TRUE)

names(Dental)<-c("MEASURE", "SEX", "AGE", "ID")
```

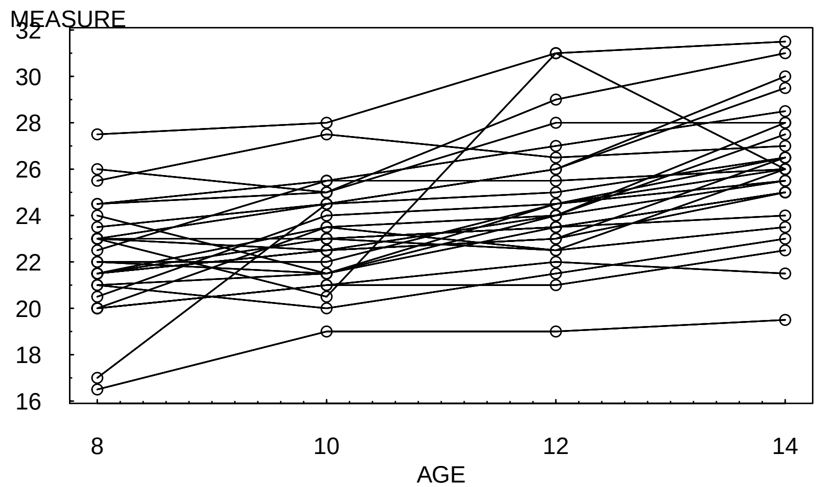
5.2 Example 5.2: Dental Data (Page 175)

This example is originally due to Potthoff and Roy (1964B); see also Rao (1987B). Here, y is the distance, measured in millimeters, from the center of the pituitary to the pteryomaxillary fissure. Measurements were taken on eleven girls and sixteen boys at ages 8, 10, 12, and 14. Of interest is the relation between the distance and age, specifically, in how the distance grows with age and whether there is a difference between males and females.

5.2.1 Figure 5.1. Multiple time series plot

```
plot(MEASURE ~ AGE, data = Dental, xlab="", ylab="", xaxt="n", yaxt="n")
 for (i in Dental$ID) {
 lines(MEASURE ~ AGE, data = subset(Dental, ID == i)) }

axis(2, at=seq(16, 32, by=2), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(16, 32, by=1), lab=F, tck=0.005)
```

```r
axis(1, at=seq(8,14, by=2), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(8,14, by=0.2), lab=F, tck=0.005)
mtext("MEASURE", side=2, line=-2, at=32.5, font=10, cex=1, las=1)
mtext("AGE", side=1, line=2, at=11, font=10, cex=1, las=1)
```



From Figure 5.1, we can see that the measurement length grows as each child ages, although it is difficult to detect differences between boys and girls. In Figure 5.1, we use open circular plotting symbols for girls and filled circular plotting symbols for boys. Figure 5.1 does show that the ninth boy has an unusual growth pattern; this pattern can also be seen in Table 5.1.

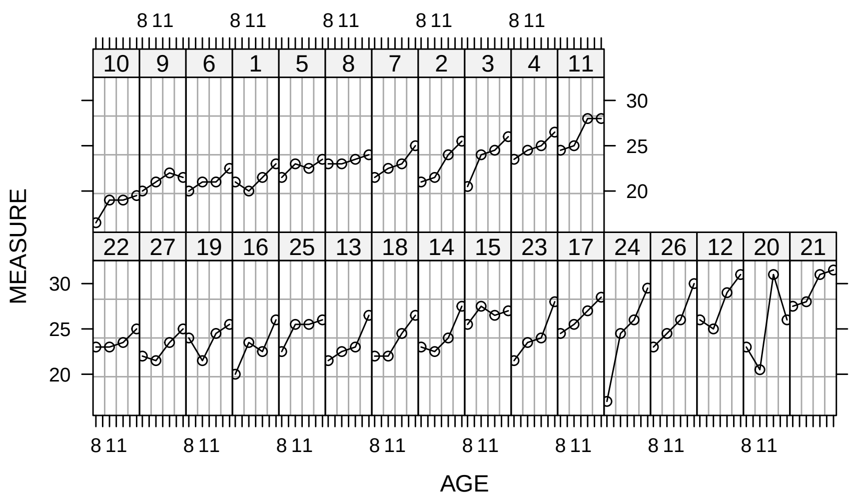
5.2.2 Summary statistics

```r
summary(Dental[, c("MEASURE")])
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  16.50   22.00   23.75   24.02   26.00   31.50 
```

5.2.3 Trellis plot, unique in r

```r
dent1 = groupedData(MEASURE ~ AGE | ID, data=Dental, outer=~SEX)
plot(dent1, layout = c(16,2))
```



5.3 TABLE 5.2: Dental data growth-curve-model parameter estimates

5.3.1 TABLE 5.2: Error components model

```
dental1.lme<-lme(MEASURE~AGE*SEX, data=Dental, random=~1|ID)
summary(dental1.lme)
```

```
Linear mixed-effects model fit by REML
 Data: Dental
       AIC      BIC    logLik
  445.7572 461.6236 -216.8786

Random effects:
 Formula: ~1 | ID
        (Intercept) Residual
StdDev:    1.816214 1.386382

Fixed effects: MEASURE ~ AGE * SEX
                Value Std.Error DF   t-value p-value
(Intercept) 16.340625 0.9813122 79 16.651810  0.0000
AGE          0.784375 0.0775011 79 10.120823  0.0000
SEX          1.032102 1.5374208 25  0.671321  0.5082
```

```
AGE:SEX     -0.304830 0.1214209 79 -2.510520  0.0141
 Correlation:
        (Intr) AGE    SEX
AGE     -0.869
SEX     -0.638  0.555
AGE:SEX  0.555 -0.638 -0.869

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.59804400 -0.45461690  0.01578365  0.50244658  3.68620792

Number of Observations: 108
Number of Groups: 27
```

5.3.2 TABLE 5.2: Growth curve model

```
dental2.lme<-lme(MEASURE~AGE*SEX, data=Dental, random=~1+AGE|ID, correlation=corSymm(fo
#corSymm gives a general correlation structure in lme
dental2.lme
```

```
Linear mixed-effects model fit by REML
  Data: Dental
  Log-restricted-likelihood: -213.0644
  Fixed: MEASURE ~ AGE * SEX
(Intercept)         AGE         SEX     AGE:SEX
 15.9304961   0.8243798   1.4779148  -0.3483069

Random effects:
 Formula: ~1 + AGE | ID
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev     Corr
(Intercept) 1.73852535 (Intr)
AGE         0.07167425 -0.238
Residual    1.49360182

Correlation Structure: General
 Formula: ~1 | ID
 Parameter estimate(s):
 Correlation:
  1      2      3
2  0.015
3  0.172  0.017
4 -0.111  0.431  0.341
Number of Observations: 108
Number of Groups: 27
```

5.3.3 TABLE 5.2: Growth curve model - omitting 9th boy

```
Dental2<-subset(Dental, ID!=20)
dental3.lme<-update(dental2.lme, data=Dental2)
dental3.lme
```

```
Linear mixed-effects model fit by REML
  Data: Dental2
  Log-restricted-likelihood: -188.7711
  Fixed: MEASURE ~ AGE * SEX
(Intercept)         AGE         SEX     AGE:SEX
 16.8586091   0.7699492   0.6119536  -0.2975491

Random effects:
 Formula: ~1 + AGE | ID
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev     Corr
(Intercept) 1.63375372 (Intr)
AGE         0.06425145 0.028
Residual    1.28363690

Correlation Structure: General
 Formula: ~1 | ID
 Parameter estimate(s):
 Correlation:
  1      2      3
2 -0.200
3  0.080  0.518
4 -0.511  0.169  0.562
Number of Observations: 104
Number of Groups: 26
```

Table 5.2 shows the parameter estimates for this model. Here, we see that the coefficient associated with linear growth is statistically significant, over all models. Moreover, the rate of increase for girls is lower than for boys. The estimated covariance between α_{0i} and α_{1i} (which is also the estimated covariance between β_{0i} and $_{1i}$ turns out to be negative. One interpretation of the negative covariance between initial status and growth rate is that subjects who start at a low level tend to grow more quickly than those who start at higher levels, and vice versa.

For comparison purposes, Table 5.2 shows the parameter estimates with the ninth boy deleted. The effects of this subject deletion on the parameter estimates are small. Table 5.2 also shows parameter estimates of the errorcomponents model. This model employs the same level-1 model but with level-2 models

$$\beta_{0i} = \beta_{00} + \beta_{01}\text{GENDER}_i + \alpha_{0i}$$

$$\beta_{1i} = \beta_{10} + \beta_{11}\text{GENDER}_i$$

With parameter estimates calculated using the full data set, there again is little change in the parameter estimates. Because the results appear to be robust to both unusual subjects and model selection, we have greater confidence in our interpretations.

Chapter 6

Modeling Issues

6.1 Import Data

```
taxprep=read.table("TXTData/TaxPrep.txt", sep ="\t", quote = "",header=TRUE)

#taxprep=read.table(choose.files(), header=TRUE, sep="\t")
```

Data for this study are from the Statistics of Income (SOI) Panel of Individual Returns, a part of the Ernst and Young/University of Michigan Tax Research Database. The SOI Panel represents a simple random sample of unaudited individual income tax returns filed for tax years 1979-1990. The data are compiled from a stratified probability sample of unaudited individual income tax returns, Forms 1040, 1040A and 1040EZ, filed by U.S. taxpayers. The estimates that are obtained from these data are intended to represent all returns filed for the income tax years under review. All returns processed are subjected to sampling except tentative and amended returns.

Variable	Description
MS	is an indicator variable of the taxpayer's marital status. It is coded one if the taxpayer is married and zero otherwise.
HH	is an indicator variable, one if the taxpayer is a head of household and zero otherwise.
DEPEND	is the number of dependents claimed by the taxpayer.
AGE	is the presence of an indicator for age 65 or over.

Variable	Description
F1040A	is an indicator variable of the taxpayer's filing type. It is coded one if the taxpayer uses Form 1040A and zero otherwise.
F1040EZ	is an indicator variable of the taxpayer's filing type. It is coded one if the taxpayer uses Form 1040EZ and zero otherwise.
TPI	is the sum of all positive income line items on the return.
TXRT	is a marginal tax rate. It is computed on TPI less exemptions and the standard deduction.
MR	is an exogenous marginal tax rate. It is computed on TPI less exemptions and the standard deduction.
EMP	is an indicator variable, one if Schedule C or F is present and zero otherwise. Self-employed taxpayers have greater need for professional assistance to reduce the reporting risks of doing business.
PREP	is a variable indicating the presence of a paid preparer.
TAX	is the tax liability on the return.
SUBJECT	Subject identifier, 1-258.
TIME	Time identifier, 1-5.
LNTAX	is the natural logarithm of the tax liability on the return.
LNTPI	is the natural logarithm of the sum of all positive income line items on the return.

6.2 Example 7.2: Income Tax Payments (Page 248)

To illustrate the performance of the fixed-effects estimators and omitted-variable tests, we examine data on determinants of income tax payments introduced in Section 3.2. Specifically, we begin with the error-components model with K = 8 coefficients estimated using generalized least squares.

6.2.1 TABLE 7.1: Fixed effects estimators

```
taxprep$YEAR<-taxprep$TIME+1981
taxprep$SUBFACTOR<-factor(taxprep$SUBJECT)
library(nlme)
taxprepfx<-lm(LNTAX~MS+HH+AGE+EMP+PREP+LNTPI+DEPEND+MR+SUBFACTOR-1, data=taxprep)
summary(taxprepfx)
```

```
Call:
lm(formula = LNTAX ~ MS + HH + AGE + EMP + PREP + LNTPI + DEPEND +
    MR + SUBFACTOR - 1, data = taxprep)

Residuals:
    Min      1Q  Median      3Q     Max
-7.4350 -0.3315 -0.0078  0.4586  6.9348

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
MS            0.072328   0.255221   0.283 0.776933
HH           -0.706799   0.326079  -2.168 0.030421 *
AGE           0.001840   0.322918   0.006 0.995456
EMP          -0.244247   0.247434  -0.987 0.323817
PREP         -0.029685   0.163207  -0.182 0.855707
LNTPI         0.716755   0.077101   9.296  < 2e-16 ***
DEPEND       -0.069021   0.082707  -0.835 0.404184
MR            0.121920   0.008998  13.550  < 2e-16 ***
SUBFACTOR1   -1.941454   0.912856  -2.127 0.033676 *
SUBFACTOR2   -2.076922   0.921470  -2.254 0.024412 *
SUBFACTOR3   -3.762761   0.867812  -4.336 1.59e-05 ***
SUBFACTOR4   -2.390221   0.936929  -2.551 0.010882 *
SUBFACTOR5   -2.383235   0.913485  -2.609 0.009214 **
SUBFACTOR6   -3.442848   0.972091  -3.542 0.000415 ***
SUBFACTOR7   -2.396985   1.026946  -2.334 0.019784 *
SUBFACTOR8   -3.901584   0.984147  -3.964 7.87e-05 ***
SUBFACTOR9   -1.792381   0.935780  -1.915 0.055721 .
SUBFACTOR10  -1.733623   0.887827  -1.953 0.051132 .
```

```
SUBFACTOR11  -2.175789   0.896572  -2.427 0.015405 *
SUBFACTOR12  -2.884418   0.692702  -4.164 3.39e-05 ***
SUBFACTOR13  -2.124878   0.974428  -2.181 0.029437 *
SUBFACTOR14  -2.489158   0.970216  -2.566 0.010442 *
SUBFACTOR15  -0.886070   0.950740  -0.932 0.351566
SUBFACTOR16  -1.903056   0.902355  -2.109 0.035188 *
SUBFACTOR17  -3.103433   0.948772  -3.271 0.001107 **
SUBFACTOR18  -7.007031   0.976968  -7.172 1.41e-12 ***
SUBFACTOR19  -2.441594   0.948031  -2.575 0.010151 *
SUBFACTOR20  -3.898509   1.028651  -3.790 0.000159 ***
SUBFACTOR21  -3.325560   0.930155  -3.575 0.000366 ***
SUBFACTOR22  -2.071372   0.891475  -2.324 0.020346 *
SUBFACTOR23  -2.350709   0.935508  -2.513 0.012132 *
SUBFACTOR24  -2.066505   0.900165  -2.296 0.021895 *
SUBFACTOR25  -5.681510   0.909637  -6.246 6.17e-10 ***
SUBFACTOR26  -4.114085   0.998612  -4.120 4.10e-05 ***
SUBFACTOR27  -1.895995   0.914310  -2.074 0.038358 *
SUBFACTOR28  -6.776403   0.929486  -7.290 6.17e-13 ***
SUBFACTOR29  -1.979414   0.892364  -2.218 0.026762 *
SUBFACTOR30  -2.253438   0.877853  -2.567 0.010400 *
SUBFACTOR31  -3.109170   0.922367  -3.371 0.000777 ***
SUBFACTOR32  -1.644017   0.934853  -1.759 0.078947 .
SUBFACTOR33  -3.595152   0.880644  -4.082 4.80e-05 ***
SUBFACTOR34  -1.282029   0.868010  -1.477 0.139990
SUBFACTOR35  -1.981843   0.981292  -2.020 0.043682 *
SUBFACTOR36  -3.176758   0.956270  -3.322 0.000925 ***
SUBFACTOR37  -2.881841   0.941031  -3.062 0.002253 **
SUBFACTOR38  -2.037214   0.912517  -2.233 0.025796 *
SUBFACTOR39  -2.490816   0.963816  -2.584 0.009894 **
SUBFACTOR40  -2.021895   0.898985  -2.249 0.024719 *
SUBFACTOR41  -2.514656   0.903545  -2.783 0.005483 **
SUBFACTOR42  -3.547532   0.995653  -3.563 0.000384 ***
SUBFACTOR43  -1.665460   0.942714  -1.767 0.077582 .
SUBFACTOR44  -1.652095   0.914844  -1.806 0.071231 .
SUBFACTOR45  -3.561106   0.950161  -3.748 0.000188 ***
SUBFACTOR46  -2.990858   0.952594  -3.140 0.001740 **
SUBFACTOR47  -2.324781   0.961738  -2.417 0.015811 *
SUBFACTOR48  -2.006750   0.754964  -2.658 0.007981 **
SUBFACTOR49  -2.597448   0.926920  -2.802 0.005171 **
SUBFACTOR50  -3.654927   1.016935  -3.594 0.000341 ***
SUBFACTOR51  -2.202546   0.897783  -2.453 0.014320 *
SUBFACTOR52  -2.796828   0.928213  -3.013 0.002649 **
SUBFACTOR53  -2.152217   0.956570  -2.250 0.024665 *
SUBFACTOR54  -2.381863   0.905095  -2.632 0.008626 **
SUBFACTOR55  -1.922384   0.913789  -2.104 0.035644 *
SUBFACTOR56  -1.156258   0.937277  -1.234 0.217622
```

```
SUBFACTOR57  -3.639612   0.953761  -3.816 0.000144 ***
SUBFACTOR58  -1.941540   0.915812  -2.120 0.034244 *
SUBFACTOR59  -1.269146   0.931013  -1.363 0.173123
SUBFACTOR60  -3.086963   0.894974  -3.449 0.000585 ***
SUBFACTOR61  -2.158203   0.896433  -2.408 0.016236 *
SUBFACTOR62  -2.767490   0.906454  -3.053 0.002323 **
SUBFACTOR63  -3.067190   0.941421  -3.258 0.001159 **
SUBFACTOR64  -3.209717   0.915086  -3.508 0.000472 ***
SUBFACTOR65  -3.936287   0.949585  -4.145 3.67e-05 ***
SUBFACTOR66  -1.657242   0.893698  -1.854 0.063974 .
SUBFACTOR67  -3.618607   0.975472  -3.710 0.000219 ***
SUBFACTOR68  -3.442074   1.006949  -3.418 0.000655 ***
SUBFACTOR69  -1.863437   0.892102  -2.089 0.036971 *
SUBFACTOR70  -2.025643   0.962731  -2.104 0.035617 *
SUBFACTOR71  -2.070916   0.909826  -2.276 0.023042 *
SUBFACTOR72  -3.560836   0.933093  -3.816 0.000144 ***
SUBFACTOR73  -1.956272   0.883031  -2.215 0.026952 *
SUBFACTOR74  -2.511433   0.942049  -2.666 0.007799 **
SUBFACTOR75  -1.548801   0.915574  -1.692 0.091023 .
SUBFACTOR76  -1.811015   0.925309  -1.957 0.050595 .
SUBFACTOR77  -1.621423   0.904550  -1.793 0.073345 .
SUBFACTOR78  -1.673650   0.905861  -1.848 0.064951 .
SUBFACTOR79  -5.856583   0.899390  -6.512 1.16e-10 ***
SUBFACTOR80  -3.704689   0.898899  -4.121 4.07e-05 ***
SUBFACTOR81  -3.322793   0.931023  -3.569 0.000375 ***
SUBFACTOR82  -1.864121   0.957077  -1.948 0.051721 .
SUBFACTOR83  -5.491182   0.961071  -5.714 1.45e-08 ***
SUBFACTOR84  -2.609013   0.941254  -2.772 0.005675 **
SUBFACTOR85  -5.323047   0.879880  -6.050 2.03e-09 ***
SUBFACTOR86  -2.829677   0.949784  -2.979 0.002957 **
SUBFACTOR87  -3.703492   0.964595  -3.839 0.000131 ***
SUBFACTOR88  -4.818659   1.016989  -4.738 2.46e-06 ***
SUBFACTOR89  -3.394560   0.930317  -3.649 0.000277 ***
SUBFACTOR90  -1.532264   0.896465  -1.709 0.087711 .
SUBFACTOR91  -1.801299   0.882717  -2.041 0.041544 *
SUBFACTOR92  -8.219328   0.888945  -9.246  < 2e-16 ***
SUBFACTOR93  -2.407979   0.912390  -2.639 0.008436 **
SUBFACTOR94  -2.845610   1.017056  -2.798 0.005240 **
SUBFACTOR95  -2.031485   0.958790  -2.119 0.034348 *
SUBFACTOR96  -2.702229   0.952599  -2.837 0.004648 **
SUBFACTOR97  -5.384899   0.905033  -5.950 3.68e-09 ***
SUBFACTOR98  -2.131225   0.924700  -2.305 0.021379 *
SUBFACTOR99  -2.625805   0.947271  -2.772 0.005673 **
SUBFACTOR100 -2.172483   0.972282  -2.234 0.025671 *
SUBFACTOR101 -2.890329   0.983665  -2.938 0.003374 **
SUBFACTOR102 -3.918986   0.870792  -4.500 7.56e-06 ***
```

```
SUBFACTOR103 -1.823848   0.910516  -2.003 0.045431 *
SUBFACTOR104 -2.140979   0.879129  -2.435 0.015048 *
SUBFACTOR105 -2.452705   0.902278  -2.718 0.006672 **
SUBFACTOR106 -2.018929   0.899036  -2.246 0.024938 *
SUBFACTOR107 -3.278959   0.904614  -3.625 0.000304 ***
SUBFACTOR108 -3.951069   0.876380  -4.508 7.29e-06 ***
SUBFACTOR109 -2.577744   0.932250  -2.765 0.005793 **
SUBFACTOR110 -3.002542   0.934017  -3.215 0.001347 **
SUBFACTOR111 -1.118914   0.953960  -1.173 0.241103
SUBFACTOR112 -2.769722   0.939232  -2.949 0.003261 **
SUBFACTOR113 -2.308694   0.913965  -2.526 0.011686 *
SUBFACTOR114 -2.596360   0.928304  -2.797 0.005256 **
SUBFACTOR115 -2.524912   0.957479  -2.637 0.008490 **
SUBFACTOR116 -1.070564   0.964510  -1.110 0.267279
SUBFACTOR117 -2.981548   0.914100  -3.262 0.001144 **
SUBFACTOR118 -2.898291   0.895760  -3.236 0.001253 **
SUBFACTOR119 -1.678321   0.927011  -1.810 0.070517 .
SUBFACTOR120 -3.646692   0.991089  -3.679 0.000246 ***
SUBFACTOR121 -2.360121   0.948188  -2.489 0.012965 *
SUBFACTOR122 -4.301704   0.961525  -4.474 8.54e-06 ***
SUBFACTOR123 -2.321742   0.936552  -2.479 0.013334 *
SUBFACTOR124 -1.885206   0.912533  -2.066 0.039089 *
SUBFACTOR125 -2.760263   0.956213  -2.887 0.003975 **
SUBFACTOR126 -4.599824   0.910184  -5.054 5.13e-07 ***
SUBFACTOR127 -1.495260   0.994569  -1.503 0.133038
SUBFACTOR128 -1.587560   0.943411  -1.683 0.092721 .
SUBFACTOR129 -2.249726   0.941817  -2.389 0.017088 *
SUBFACTOR130 -2.513272   0.931073  -2.699 0.007062 **
SUBFACTOR131 -2.914927   0.902195  -3.231 0.001273 **
SUBFACTOR132 -1.912501   0.895668  -2.135 0.032975 *
SUBFACTOR133 -2.844954   0.883279  -3.221 0.001318 **
SUBFACTOR134 -2.486082   0.961257  -2.586 0.009839 **
SUBFACTOR135 -1.782512   0.945921  -1.884 0.059791 .
SUBFACTOR136 -3.321478   0.959246  -3.463 0.000557 ***
SUBFACTOR137 -1.364910   0.949280  -1.438 0.150786
SUBFACTOR138 -2.180505   0.975733  -2.235 0.025650 *
SUBFACTOR139 -6.851310   0.964615  -7.103 2.29e-12 ***
SUBFACTOR140 -3.264175   0.961476  -3.395 0.000713 ***
SUBFACTOR141 -3.277959   0.925814  -3.541 0.000417 ***
SUBFACTOR142 -2.047689   0.878153  -2.332 0.019904 *
SUBFACTOR143 -3.311763   0.999429  -3.314 0.000953 ***
SUBFACTOR144 -3.224253   0.882052  -3.655 0.000270 ***
SUBFACTOR145 -1.602488   0.945951  -1.694 0.090560 .
SUBFACTOR146 -3.433803   0.919533  -3.734 0.000199 ***
SUBFACTOR147 -1.962344   0.917847  -2.138 0.032754 *
SUBFACTOR148 -5.720274   0.846794  -6.755 2.39e-11 ***
```

```
SUBFACTOR149 -2.394029   0.935963  -2.558 0.010676 *
SUBFACTOR150 -2.313197   0.913255  -2.533 0.011460 *
SUBFACTOR151 -2.661345   1.004407  -2.650 0.008181 **
SUBFACTOR152 -2.874865   0.874572  -3.287 0.001046 **
SUBFACTOR153 -2.324181   0.902537  -2.575 0.010159 *
SUBFACTOR154 -2.125162   0.914003  -2.325 0.020261 *
SUBFACTOR155 -3.781776   0.951065  -3.976 7.49e-05 ***
SUBFACTOR156 -3.755601   0.944757  -3.975 7.53e-05 ***
SUBFACTOR157 -4.081932   0.937647  -4.353 1.48e-05 ***
SUBFACTOR158 -6.112004   0.942740  -6.483 1.39e-10 ***
SUBFACTOR159 -3.983963   0.989367  -4.027 6.07e-05 ***
SUBFACTOR160 -2.913340   0.921931  -3.160 0.001624 **
SUBFACTOR161 -2.042601   0.975596  -2.094 0.036532 *
SUBFACTOR162 -3.397019   0.971739  -3.496 0.000493 ***
SUBFACTOR163 -1.617177   0.917376  -1.763 0.078228 .
SUBFACTOR164 -2.630423   0.889036  -2.959 0.003160 **
SUBFACTOR165 -4.185708   0.893828  -4.683 3.21e-06 ***
SUBFACTOR166 -2.434348   0.917949  -2.652 0.008127 **
SUBFACTOR167 -1.390578   0.974019  -1.428 0.153692
SUBFACTOR168 -4.853027   0.957698  -5.067 4.78e-07 ***
SUBFACTOR169 -2.283081   0.923557  -2.472 0.013596 *
SUBFACTOR170 -4.372778   0.959421  -4.558 5.79e-06 ***
SUBFACTOR171 -3.425975   0.902129  -3.798 0.000155 ***
SUBFACTOR172 -2.343538   0.920833  -2.545 0.011073 *
SUBFACTOR173 -1.710324   0.910104  -1.879 0.060492 .
SUBFACTOR174 -2.098796   0.954269  -2.199 0.028074 *
SUBFACTOR175 -2.797872   0.913217  -3.064 0.002243 **
SUBFACTOR176 -5.046590   0.857151  -5.888 5.31e-09 ***
SUBFACTOR177 -2.893347   0.921484  -3.140 0.001739 **
SUBFACTOR178 -1.841189   0.887291  -2.075 0.038229 *
SUBFACTOR179 -4.466157   0.966026  -4.623 4.26e-06 ***
SUBFACTOR180 -3.730520   0.920546  -4.053 5.45e-05 ***
SUBFACTOR181 -2.869046   0.931615  -3.080 0.002128 **
SUBFACTOR182 -2.424206   0.887707  -2.731 0.006425 **
SUBFACTOR183 -5.356722   0.936315  -5.721 1.39e-08 ***
SUBFACTOR184 -3.066164   0.942504  -3.253 0.001178 **
SUBFACTOR185 -5.124591   0.908341  -5.642 2.18e-08 ***
SUBFACTOR186 -3.251203   0.991743  -3.278 0.001080 **
SUBFACTOR187 -1.677176   0.902537  -1.858 0.063414 .
SUBFACTOR188 -3.472789   0.937982  -3.702 0.000225 ***
SUBFACTOR189 -3.762196   0.964176  -3.902 0.000102 ***
SUBFACTOR190 -2.219572   0.810106  -2.740 0.006253 **
SUBFACTOR191 -2.800552   0.908851  -3.081 0.002115 **
SUBFACTOR192 -3.399641   0.949269  -3.581 0.000358 ***
SUBFACTOR193 -2.837433   0.950723  -2.985 0.002908 **
SUBFACTOR194 -3.019642   0.910231  -3.317 0.000940 ***
```

```
SUBFACTOR195 -2.440036   0.937527  -2.603 0.009385 **
SUBFACTOR196 -3.858337   0.947051  -4.074 4.98e-05 ***
SUBFACTOR197 -2.864903   0.978925  -2.927 0.003503 **
SUBFACTOR198 -2.397067   0.987467  -2.427 0.015375 *
SUBFACTOR199 -0.967048   0.997075  -0.970 0.332333
SUBFACTOR200 -3.281440   0.937895  -3.499 0.000488 ***
SUBFACTOR201 -2.309235   0.984349  -2.346 0.019169 *
SUBFACTOR202 -1.779309   0.803265  -2.215 0.026973 *
SUBFACTOR203 -2.595728   0.883891  -2.937 0.003391 **
SUBFACTOR204 -1.802010   0.915415  -1.969 0.049278 *
SUBFACTOR205 -2.116093   0.987569  -2.143 0.032370 *
SUBFACTOR206 -1.809473   0.920028  -1.967 0.049481 *
SUBFACTOR207 -1.560251   0.954835  -1.634 0.102555
SUBFACTOR208 -1.883087   0.892272  -2.110 0.035062 *
SUBFACTOR209 -3.478732   0.939333  -3.703 0.000224 ***
SUBFACTOR210 -3.147438   0.962608  -3.270 0.001112 **
SUBFACTOR211 -2.757256   0.910277  -3.029 0.002515 **
SUBFACTOR212 -1.672145   0.935748  -1.787 0.074239 .
SUBFACTOR213 -2.927508   0.941279  -3.110 0.001922 **
SUBFACTOR214 -3.097024   0.950635  -3.258 0.001159 **
SUBFACTOR215 -2.887754   0.940748  -3.070 0.002200 **
SUBFACTOR216 -1.979758   1.017965  -1.945 0.052070 .
SUBFACTOR217 -2.785545   0.954274  -2.919 0.003588 **
SUBFACTOR218 -4.731554   0.954436  -4.957 8.36e-07 ***
SUBFACTOR219 -4.117183   0.992160  -4.150 3.61e-05 ***
SUBFACTOR220 -3.334648   0.952678  -3.500 0.000485 ***
SUBFACTOR221 -3.477129   0.971094  -3.581 0.000359 ***
SUBFACTOR222 -4.151081   0.821387  -5.054 5.13e-07 ***
SUBFACTOR223 -2.232397   0.882094  -2.531 0.011529 *
SUBFACTOR224 -2.616304   0.866332  -3.020 0.002591 **
SUBFACTOR225 -1.940628   0.875393  -2.217 0.026851 *
SUBFACTOR226 -2.011574   0.908631  -2.214 0.027059 *
SUBFACTOR227 -2.430288   0.908411  -2.675 0.007585 **
SUBFACTOR228 -2.102822   0.903471  -2.327 0.020133 *
SUBFACTOR229 -3.447302   0.927645  -3.716 0.000213 ***
SUBFACTOR230 -2.344091   0.912392  -2.569 0.010335 *
SUBFACTOR231 -3.459879   0.935278  -3.699 0.000228 ***
SUBFACTOR232 -5.658765   1.016771  -5.565 3.34e-08 ***
SUBFACTOR233 -4.783141   0.925671  -5.167 2.86e-07 ***
SUBFACTOR234 -3.819151   0.856194  -4.461 9.08e-06 ***
SUBFACTOR235 -2.024762   0.949219  -2.133 0.033155 *
SUBFACTOR236 -2.784329   0.910186  -3.059 0.002278 **
SUBFACTOR237 -3.198397   0.944980  -3.385 0.000740 ***
SUBFACTOR238 -3.142874   0.919383  -3.418 0.000655 ***
SUBFACTOR239 -3.439833   0.940339  -3.658 0.000267 ***
SUBFACTOR240 -2.622761   0.989240  -2.651 0.008142 **
```

```
SUBFACTOR241 -3.996097   0.871946  -4.583 5.15e-06 ***
SUBFACTOR242 -5.086598   0.965775  -5.267 1.69e-07 ***
SUBFACTOR243 -2.900497   0.930985  -3.116 0.001887 **
SUBFACTOR244 -1.575051   0.894947  -1.760 0.078717 .
SUBFACTOR245 -2.699959   0.941336  -2.868 0.004213 **
SUBFACTOR246 -3.595091   0.939563  -3.826 0.000138 ***
SUBFACTOR247 -1.807229   0.982754  -1.839 0.066213 .
SUBFACTOR248 -3.003435   0.930749  -3.227 0.001291 **
SUBFACTOR249 -4.050990   0.958601  -4.226 2.59e-05 ***
SUBFACTOR250 -3.054127   0.939440  -3.251 0.001187 **
SUBFACTOR251 -2.856217   0.899253  -3.176 0.001537 **
SUBFACTOR252 -2.139357   0.886316  -2.414 0.015963 *
SUBFACTOR253 -1.074312   0.963784  -1.115 0.265249
SUBFACTOR254 -2.605768   1.015898  -2.565 0.010459 *
SUBFACTOR255 -1.831341   0.927795  -1.974 0.048666 *
SUBFACTOR256 -1.873042   0.951552  -1.968 0.049291 *
SUBFACTOR257 -1.409751   0.951357  -1.482 0.138693
SUBFACTOR258 -0.117362   0.884154  -0.133 0.894426
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.373 on 1024 degrees of freedom
Multiple R-squared:  0.9726,    Adjusted R-squared:  0.9655
F-statistic: 136.6 on 266 and 1024 DF,  p-value: < 2.2e-16
```

6.2.2 TABLE 7.1: Random effects estimator

```
taxpreprdm1<-lme(LNTAX~MS+HH+AGE+EMP+PREP+LNTPI+DEPEND+MR, data=taxprep, random=~1|SUBJECT, metho
summary(taxpreprdm1)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: taxprep
       AIC      BIC    logLik
  4813.255 4870.041 -2395.627

Random effects:
 Formula: ~1 | SUBJECT
        (Intercept) Residual
StdDev:   0.9602161 1.368896

Fixed effects: LNTAX ~ MS + HH + AGE + EMP + PREP + LNTPI + DEPEND + MR
                 Value Std.Error   DF   t-value p-value
(Intercept) -2.9603371 0.5705536 1024 -5.188534  0.0000
MS           0.0373000 0.1824839 1024  0.204402  0.8381
HH          -0.6889876 0.2320057 1024 -2.969702  0.0031
```

```
AGE          0.0207431 0.2000035 1024  0.103713  0.9174
EMP         -0.5048035 0.1679848 1024 -3.005054  0.0027
PREP        -0.0217036 0.1175229 1024 -0.184675  0.8535
LNTPI        0.7604058 0.0699692 1024 10.867728  0.0000
DEPEND      -0.1127475 0.0592818 1024 -1.901891  0.0575
MR           0.1153752 0.0073142 1024 15.774213  0.0000
 Correlation:
       (Intr) MS     HH     AGE    EMP    PREP   LNTPI  DEPEND
MS      0.176
HH      0.030  0.419
AGE    -0.043 -0.167 -0.023
EMP    -0.116 -0.069  0.024 -0.030
PREP   -0.035 -0.045  0.004 -0.115 -0.112
LNTPI  -0.948 -0.180 -0.081 -0.043  0.099 -0.016
DEPEND -0.074 -0.604 -0.269  0.224 -0.038 -0.039 -0.068
MR      0.522 -0.020  0.055  0.149 -0.041 -0.051 -0.698  0.102

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-5.83483692 -0.21263981  0.09677632  0.39814646  5.79731648

Number of Observations: 1290
Number of Groups: 258
```

6.2.3 Hausman's test

```
beta1fix<-coefficients(taxprepfx)
beta1fe<-beta1fix[1:8]
cov1fix<-vcov(taxprepfx)
cov1fe<-cov1fix[1:8, 1:8]
beta1re<-coefficients(taxpreprdm1)
beta1re<-t(beta1re[1, 2:9])
cov1re<-vcov(taxpreprdm1)
cov1re<-cov1re[2:9, 2:9]
HSTEST1<-t(beta1fe-beta1re)%*%solve(cov1fe-cov1re)%*%(beta1fe-beta1re)
beta1fe
```

```
          MS           HH          AGE          EMP         PREP
 0.072327932 -0.706799308  0.001839538 -0.244247153 -0.029685211
       LNTPI       DEPEND           MR
 0.716754955 -0.069020879  0.121919964
```

```
beta1re
```

```
                 1
MS      0.03730005
HH     -0.68898764
```

```
AGE     0.02074305
EMP    -0.50480349
PREP   -0.02170360
LNTPI   0.76040578
DEPEND -0.11274746
MR      0.11537523
```

```
HSTEST1
```

```
         1
1 6.019006
```

6.3 Example 7.2: Income Tax Payments (continued) (Page 255)

6.3.1 Table 7.2: Fixed effects estimators with two variable slopes

```
ACF(taxpreprdm1, maxlag=10) #Obtain ACF of residuals for within-group residual
```

```
  lag          ACF
1   0  1.000000000
2   1 -0.004283774
3   2 -0.223519705
4   3 -0.307380297
5   4 -0.355268841
```

```
# Compared with SAS, lm in R can estimate fixed effects, but can not code AR(1) for within-subjec
taxprepfx2<-lm(LNTAX~MS+HH+AGE+EMP+PREP+LNTPI+DEPEND+MR+SUBFACTOR+SUBFACTOR*MR+SUBFACTOR*LNTPI-1,
# summary(taxprepfx2)
```

6.3.2 Table 7.2: Variable slopes model

```
taxpreprdm2<-lme(LNTAX~MS+HH+AGE+EMP+PREP+LNTPI+DEPEND+MR, data=taxprep, method="ML",random=~1+LN
# I changed the initial code to "control = lmeControl(opt = "optim")", because the initial code h

summary(taxpreprdm2) #ESTIMATES ARE CLOSE TO RESULTS FROM SAS
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: taxprep
       AIC      BIC    logLik
  4443.141 4530.902 -2204.571

Random effects:
 Formula: ~1 + LNTPI + MR | SUBJECT
```

```
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev      Corr
(Intercept) 12.05966691 (Intr) LNTPI
LNTPI        1.27245273 -0.988
MR           0.07050666  0.475 -0.602
Residual     1.14826017

Correlation Structure: AR(1)
 Formula: ~1 | SUBJECT
 Parameter estimate(s):
      Phi
0.1346485
Fixed effects: LNTAX ~ MS + HH + AGE + EMP + PREP + LNTPI + DEPEND + MR
                 Value Std.Error   DF   t-value p-value
(Intercept) -14.560716 1.4762035 1024 -9.863624  0.0000
MS           -0.613181 0.1607932 1024 -3.813475  0.0001
HH           -0.766651 0.1991612 1024 -3.849398  0.0001
AGE          -0.372122 0.1711989 1024 -2.173622  0.0300
EMP          -0.646505 0.1346603 1024 -4.801007  0.0000
PREP         -0.303705 0.0960482 1024 -3.162005  0.0016
LNTPI         2.268717 0.1693620 1024 13.395665  0.0000
DEPEND       -0.140338 0.0495257 1024 -2.833637  0.0047
MR            0.006456 0.0102326 1024  0.630904  0.5282
 Correlation:
       (Intr) MS     HH     AGE    EMP    PREP   LNTPI  DEPEND
MS      0.293
HH      0.070  0.450
AGE    -0.011 -0.139 -0.001
EMP    -0.009 -0.051  0.016 -0.053
PREP    0.053 -0.019  0.012 -0.118 -0.085
LNTPI  -0.990 -0.303 -0.095 -0.021  0.002 -0.071
DEPEND  0.044 -0.549 -0.250  0.235 -0.030 -0.037 -0.094
MR      0.733  0.181  0.098  0.099  0.011  0.027 -0.808  0.128

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-7.2788124 -0.1668237  0.0753182  0.3376614  2.7774163

Number of Observations: 1290
Number of Groups: 258
```

6.3.3 Hausman's test

```
beta2fix<-coefficients(taxprepfx2)
beta2fe<-beta2fix[1:8]
```

```
cov2fix<-vcov(taxprepfx2)
cov2fe<-cov2fix[1:8, 1:8]
beta2re<-coefficients(taxpreprdm2)
beta2re<-t(beta2re[1, 2:9])
cov2re<-vcov(taxpreprdm2)
cov2re<-cov2re[2:9, 2:9]
HSTEST2<-t(beta2fe-beta2re)%*%solve(cov2fe-cov2re)%*%(beta2fe-beta2re)
beta2fe
```

```
         MS          HH         AGE         EMP        PREP       LNTPI
-0.28247941 -2.19247828 -0.54479788 -0.12152994 -0.47339937  0.62023798
     DEPEND          MR
-0.29578737  0.02681867
```

```
beta2re
```

```
                  1
MS     -0.613180767
HH     -0.766650688
AGE    -0.372121718
EMP    -0.646504958
PREP   -0.303704908
LNTPI   1.680299311
DEPEND -0.140337926
MR     -0.007261631
```

```
HSTEST2 #ESTIMATES ARE DIFFERENT FROM RESULTS FROM SAS, BECAUSE THE FIXED EFFECTS ESTIMATORS DID
```

```
         1
1 27.30712
```

6.4 TABLE 7.3 Augmented regressions

6.4.1 Create panel data set with subject averages

```
msavg<-aggregate(taxprep$MS, list(SUBJECT=taxprep$SUBJECT), mean)
names(msavg)<-c("SUBJECT", "msavg")
hhavg<-aggregate(taxprep$HH, list(SUBJECT=taxprep$SUBJECT), mean)
names(hhavg)<-c("SUBJECT", "hhavg")
ageavg<-aggregate(taxprep$AGE, list(SUBJECT=taxprep$SUBJECT), mean)
names(ageavg)<-c("SUBJECT", "ageavg")
empavg<-aggregate(taxprep$EMP, list(SUBJECT=taxprep$SUBJECT), mean)
names(empavg)<-c("SUBJECT", "empavg")
prepavg<-aggregate(taxprep$PREP, list(SUBJECT=taxprep$SUBJECT), mean)
names(prepavg)<-c("SUBJECT", "prepavg")
dependavg<-aggregate(taxprep$DEPEND, list(SUBJECT=taxprep$SUBJECT), mean)
```

```
names(dependavg)<-c("SUBJECT", "dependavg")
lntpiavg<-aggregate(taxprep$LNTPI, list(SUBJECT=taxprep$SUBJECT), mean)
names(lntpiavg)<-c("SUBJECT", "lntpiavg")
mravg<-aggregate(taxprep$MR, list(SUBJECT=taxprep$SUBJECT), mean)
names(mravg)<-c("SUBJECT", "mravg")

avg<-merge(msavg, taxprep, by="SUBJECT", all.y=T, sort=T)
avg<-merge(hhavg, avg, by="SUBJECT", all.y=T, sort=T)
avg<-merge(ageavg, avg, by="SUBJECT", all.y=T, sort=T)
avg<-merge(empavg, avg, by="SUBJECT", all.y=T, sort=T)
avg<-merge(prepavg, avg, by="SUBJECT", all.y=T, sort=T)
avg<-merge(dependavg, avg, by="SUBJECT", all.y=T, sort=T)
avg<-merge(lntpiavg, avg, by="SUBJECT", all.y=T, sort=T)
avg<-merge(mravg, avg, by="SUBJECT", all.y=T, sort=T)
```

6.4.2 Models with averages as omitted variables

```
#VARIABLE INTERCEPTS AND TWO VARIABLE SLOPES
taxprepaug<-lme(LNTAX~MS+HH+AGE+EMP+PREP+LNTPI+DEPEND+MR+msavg+hhavg+ageavg+empavg+prep
#Again, I change the code to "control = lmeControl(opt = "optim")" due to convergence p
summary(taxprepaug)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: avg
      AIC     BIC    logLik
  4412.59 4541.65 -2181.295

Random effects:
 Formula: ~1 + LNTPI + MR | SUBJECT
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev      Corr
(Intercept) 12.48682715 (Intr) LNTPI
LNTPI        1.28389597 -0.992
MR           0.05703604  0.447 -0.555
Residual     1.10067285

Correlation Structure: AR(1)
 Formula: ~1 | SUBJECT
 Parameter estimate(s):
        Phi
-0.04702359
Fixed effects: LNTAX ~ MS + HH + AGE + EMP + PREP + LNTPI + DEPEND + MR + msavg +
                 Value Std.Error   DF    t-value p-value
(Intercept) -22.909909 2.2231930 1024 -10.304957  0.0000
MS           -0.563113 0.2425479 1024  -2.321658  0.0204
```

```
HH          -1.089503 0.2825216 1024  -3.856353  0.0001
AGE         -0.408585 0.2792958 1024  -1.462911  0.1438
EMP         -0.395533 0.2102914 1024  -1.880881  0.0603
PREP        -0.289016 0.1414320 1024  -2.043495  0.0413
LNTPI        2.374719 0.1680609 1024  14.130110  0.0000
DEPEND      -0.174946 0.0719544 1024  -2.431338  0.0152
MR           0.030201 0.0107346 1024   2.813397  0.0050
msavg       -0.273782 0.3121956  249  -0.876955  0.3814
hhavg        0.456298 0.3823711  249   1.193338  0.2339
ageavg       0.007476 0.3370271  249   0.022184  0.9823
empavg      -0.450047 0.2598717  249  -1.731806  0.0845
prepavg      0.035089 0.1833941  249   0.191333  0.8484
dependavg   -0.006988 0.0946377  249  -0.073840  0.9412
lntpiavg     0.962655 0.1930848  249   4.985661  0.0000
mravg       -0.109881 0.0158988  249  -6.911257  0.0000
 Correlation:
          (Intr) MS     HH     AGE    EMP    PREP   LNTPI  DEPEND MR
MS         0.159
HH         0.052  0.271
AGE        0.041 -0.043 -0.013
EMP        0.014  0.013  0.005  0.014
PREP       0.056 -0.028 -0.002 -0.018 -0.049
LNTPI     -0.716 -0.228 -0.083 -0.045  0.020 -0.073
DEPEND     0.042 -0.433 -0.185  0.101 -0.030  0.012 -0.058
MR         0.423  0.136  0.075  0.130 -0.001  0.057 -0.675  0.068
msavg      0.167 -0.741 -0.195  0.071 -0.006  0.038  0.023  0.354  0.005
hhavg      0.062 -0.183 -0.743  0.046 -0.005  0.020  0.017  0.144 -0.019
ageavg    -0.024  0.053  0.015 -0.816 -0.018  0.025  0.034 -0.081 -0.104
empavg    -0.033  0.001 -0.001 -0.018 -0.807  0.045 -0.027  0.032  0.008
prepavg   -0.032  0.028  0.010  0.016  0.044 -0.774  0.040 -0.007 -0.030
dependavg  0.057  0.360  0.152 -0.086  0.020  0.003 -0.013 -0.762 -0.005
lntpiavg  -0.748 -0.021 -0.002 -0.019 -0.038 -0.016  0.083 -0.007 -0.002
mravg      0.641  0.056  0.002 -0.048  0.020  0.019 -0.114 -0.006 -0.247
          msavg  hhavg  ageavg empavg prepvg dpndvg lntpvg
MS
HH
AGE
EMP
PREP
LNTPI
DEPEND
MR
msavg
hhavg      0.378
ageavg    -0.153 -0.057
empavg    -0.038  0.005  0.001
```

```
prepavg   -0.035 -0.024 -0.095 -0.086
dependavg -0.523 -0.243  0.201 -0.036 -0.042
lntpiavg  -0.253 -0.117 -0.018  0.068  0.006 -0.102
mravg      0.130  0.094  0.113 -0.032 -0.042  0.124 -0.838

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-7.19930371 -0.17697920  0.07578962  0.32209554  3.33317410

Number of Observations: 1290
Number of Groups: 258
```

```
#ESTIMATES ARE DIFFERENT FROM SAS BECAUSE fa0(3) WAS CODED IN SAS
beta3re<-coefficients(taxprepaug)
betarand<-t(beta3re[1, 10:17])
cov3re<-vcov(taxprepaug)
cov3re<-cov3re[10:17, 10:17]
ARTEST <- t(betarand)%*%solve(cov3re)%*%betarand
betarand
```

```
                     1
msavg     -0.273781537
hhavg      0.456298030
ageavg     0.007476440
empavg    -0.450047310
prepavg    0.035089331
dependavg -0.006988012
lntpiavg   0.962655240
mravg     -0.109880536
```

```
ARTEST
```

```
         1
1 59.97999
```

Chapter 7

Dynamic Models

7.1 Import Data

```
#insbeta=read.table(choose.files(), header=TRUE, sep="\t")
library(nlme)
insbeta=read.table("TXTData/insbeta.txt", sep ="\t", quote = "",header=TRUE)

insbeta$YEAR=1995+(insbeta$Time-1)/12
```

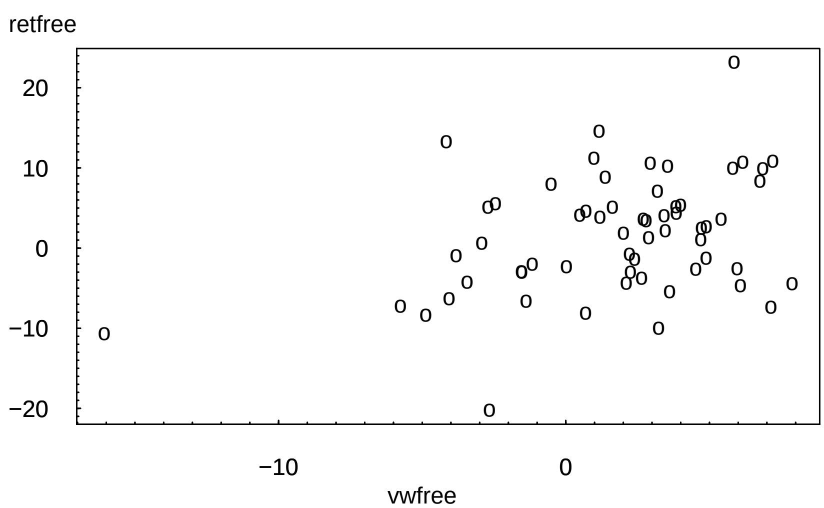
This is the data used at page 302 for 8.6 Example: Capital Asset Pricing Model. No more information could be found.

7.2 Example 8.6: Capital Asset Pricing Model (Page 302)

The capital asset pricing model (CAPM) is a representation that is widely used in financial economics. An intuitively appealing idea, and one of the basic characteristics of the CAPM, is that there should be a relationship between the performance of a security and the performance of the market. One rationale is simply that if economic forces are such that the market improves, then those same forces should act upon an individual stock, suggesting that it also improve. We measure performance of a security through the return. To measure performance of the market, several market indices exist for each exchange. As an illustration, in the following we use the return from the "value-weighted" index of the market created by the Center for Research in Securities Prices (CRSP). The value-weighted index is defined by assuming a portfolio is created when investing an amount of money in proportion to the market value (at a certain date) of firms listed on the New York Stock Exchange, the American Stock Exchange, and the Nasdaq stock market.

7.2.1 Plot of RETFREE vs. VWFREE for Incoln insurance company

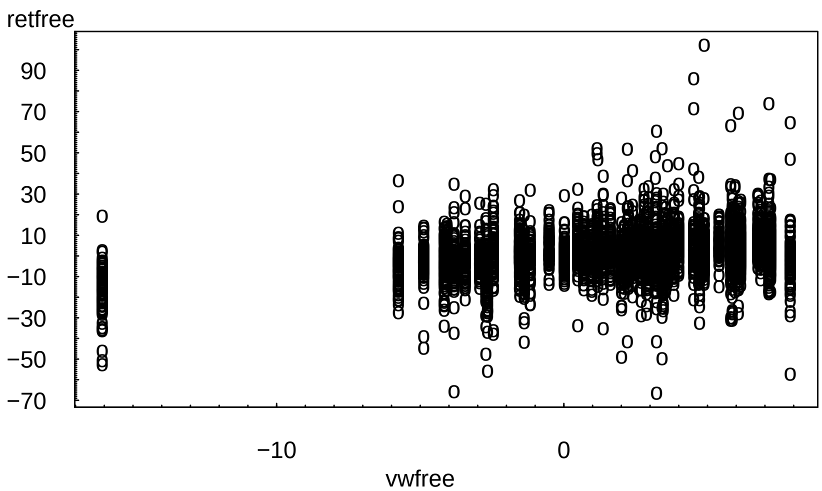
```
plot(retfree ~ vwfree, data = subset(insbeta, insbeta$PERMNO==49015), type="p", xaxt="
axis(2, at=seq(-30, 30, by=10), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(-30, 30, by=1), lab=F, tck=0.005)
axis(1, at=seq(-20,20, by=10), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(-20,20, by=1), lab=F, tck=0.005)
axis(2, at=seq(-70, 110, by=10), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(-70, 110, by=1), lab=F, tck=0.005)
axis(1, at=seq(-20,10, by=10), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(-20,10, by=1), lab=F, tck=0.005)
mtext("retfree", side=2, line=0, at=28, font=10, cex=1, las=1)
mtext("vwfree", side=1, line=2, at=-5, font=10, cex=1)
```



7.2.2 Plot of RETFREE vs. VWFREE for 90 insurance firms

```
plot(retfree ~ vwfree, data =insbeta, type="p", xaxt="n", yaxt="n", ylab="", xlab="", 
axis(2, at=seq(-70, 110, by=10), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(-70, 110, by=1), lab=F, tck=0.005)
axis(1, at=seq(-20,10, by=10), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(-20,10, by=1), lab=F, tck=0.005)
mtext("retfree", side=2, line=0, at=115, font=10, cex=1, las=1)
```

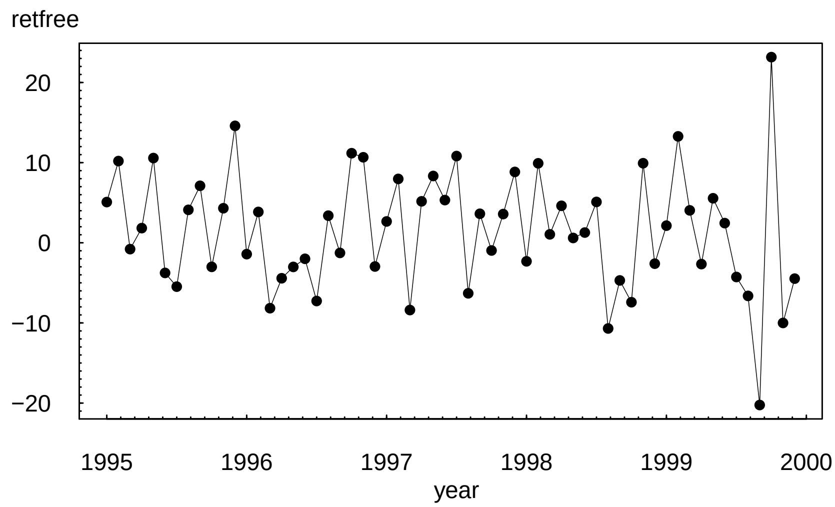
```
mtext("vwfree", side=1, line=2, at=-5, font=10, cex=1)
mtext("RETFREE vs. VWFREE for 90 Insurance Firms", side=1, line=4, at=-5, font=10, cex=1)
```



RETFREE vs. VWFREE for 90 Insurance Firms

7.2.3 Plot of RETFREE vs. YEAR for Lincoln

```
plot(retfree ~ YEAR, data = subset(insbeta, insbeta$PERMNO==49015), type="o", xaxt="n", yaxt="n",
axis(2, at=seq(-30, 30, by=10), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(-30, 30, by=1), lab=F, tck=0.005)
axis(1, at=seq(1995,2000, by=1), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(1995,2000, by=0.1), lab=F, tck=0.005)
mtext("retfree", side=2, line=0, at=28, font=10, cex=1, las=1)
mtext("year", side=1, line=2, at=1997.50, font=10, cex=1)
mtext("Lincoln RETFREE vs. YEAR", side=1, line=5, at=1997.50, font=10, cex=1)
```



7.2.4 Table 8.2 Summary statistics for market index and risk-free security

```
LINCOLN<-subset(insbeta, insbeta$PERMNO==49015)
summary(LINCOLN[, c("VWRETD", "SPRTRN", "riskf", "vwfree", "spfree")])
```

```
     VWRETD             SPRTRN              riskf            vwfree        
 Min.   :-15.6765   Min.   :-14.5797   Min.   :0.2964   Min.   :-16.0683  
 1st Qu.: -0.2581   1st Qu.:  0.1612   1st Qu.:0.3811   1st Qu.: -0.6755  
 Median :  2.9464   Median :  2.6730   Median :0.4147   Median :  2.5174  
 Mean   :  2.0914   Mean   :  2.0380   Mean   :0.4075   Mean   :  1.6839  
 3rd Qu.:  4.9429   3rd Qu.:  5.0748   3rd Qu.:0.4267   3rd Qu.:  4.5654  
 Max.   :  8.3054   Max.   :  8.0294   Max.   :0.4829   Max.   :  7.8798  
     spfree        
 Min.   :-14.9714  
 1st Qu.: -0.2533  
 Median :  2.2244  
 Mean   :  1.6305  
 3rd Qu.:  4.6481  
 Max.   :  7.7330  
```

```
sd1<-sqrt(diag(var(insbeta[,c("VWRETD", "SPRTRN", "riskf", "vwfree", "spfree")])))
sd1
```

```
    VWRETD      SPRTRN       riskf      vwfree      spfree
```

```
4.09890088 3.98794716 0.03380599 4.09997511 3.98932881
```

```
cor(LINCOLN[,c("VWRETD", "SPRTRN", "riskf", "vwfree", "spfree")])
```

```
           VWRETD      SPRTRN       riskf      vwfree      spfree
VWRETD  1.0000000  0.97950897 -0.02765660  0.99996603  0.97940410
SPRTRN  0.9795090  1.00000000 -0.03663843  0.97955443  0.99996414
riskf  -0.0276566 -0.03663843  1.00000000 -0.03589477 -0.04509984
vwfree  0.9999660  0.97955443 -0.03589477  1.00000000  0.97951935
spfree  0.9794041  0.99996414 -0.04509984  0.97951935  1.00000000
```

Table 8.2 summarizes the performance of the market through the return from the value-weighted index, VWRETD, and risk free instrument, RISKFREE. We also consider the difference between the two, VWFREE, and interpret this to be the return from the market in excess of the risk-free rate.

7.2.5 TABLE 8.3 Summary statistics for individual security returns

```
summary(insbeta[,c("RET", "retfree", "PRC")])
```

```
      RET                retfree              PRC          
 Min.   :-66.1972   Min.   :-66.5785   Min.   :    0.81  
 1st Qu.: -3.8462   1st Qu.: -4.2428   1st Qu.:   14.25  
 Median :  0.7453   Median :  0.3402   Median :   26.88  
 Mean   :  1.0521   Mean   :  0.6446   Mean   :  547.11  
 3rd Qu.:  5.8823   3rd Qu.:  5.4675   3rd Qu.:   45.89  
 Max.   :102.5000   Max.   :102.0850   Max.   :78305.00  
```

```
# STANDARD DEVIATION
sd1<-sqrt(diag(var(insbeta[,c("RET", "retfree", "PRC")])))
sd1
```

```
       RET    retfree        PRC 
  10.03772   10.03552 5178.49653 
```

```
cor(insbeta[,c("RET", "VWRETD", "SPRTRN", "riskf", "retfree", "vwfree", "spfree")])
```

```
               RET     VWRETD      SPRTRN       riskf    retfree
RET     1.00000000  0.2937725  0.28237030  0.06693926 0.99999435
VWRETD  0.29377254  1.0000000  0.97950897 -0.02765660 0.29393029
SPRTRN  0.28237030  0.9795090  1.00000000 -0.03663843 0.28255580
riskf   0.06693926 -0.0276566 -0.03663843  1.00000000 0.06358534
retfree 0.99999435  0.2939303  0.28255580  0.06358534 1.00000000
vwfree  0.29314362  0.9999660  0.97955443 -0.03589477 0.29332899
spfree  0.28170525  0.9794041  0.99996414 -0.04509984 0.28191911
             vwfree      spfree
RET      0.29314362  0.28170525
```

```
VWRETD   0.99996603  0.97940410
SPRTRN   0.97955443  0.99996414
riskf   -0.03589477 -0.04509984
retfree  0.29332899  0.28191911
vwfree   1.00000000  0.97951935
spfree   0.97951935  1.00000000
```

Table 8.3 summarizes the performance of individual securities through the monthly return, `RET`. These summary statistics are based on 5,400 monthly observations taken from 90 firms. The difference between the return and the corresponding risk-free instrument is `RETFREE`.

7.2.6 TABLE 8.4 Fixed effects models

```
#HOMOGENEOUS MODEL
insbetahomo<-gls(retfree~vwfree, method="REML", data=insbeta)
anova(insbetahomo)
```

```
Denom. DF: 5398
            numDF  F-value p-value
(Intercept)     1  24.3686  <.0001
vwfree          1 508.1788  <.0001
```

```
insbetahomo$sigma^2
```

```
[1] 92.06322
```

```
AIC(insbetahomo)
```

```
[1] 39757.19
```

```
logLik(insbetahomo)*(-2)
```

```
'log Lik.' 39751.19 (df=3)
```

```
insbeta$FACPERM<-factor(insbeta$PERMNO)
#VARIABLE INTERCEPT MODEL
insbetafx1<-gls(retfree~vwfree+FACPERM, method="REML", data=insbeta)
anova(insbetafx1)
```

```
Denom. DF: 5309
            numDF  F-value p-value
(Intercept)     1  24.2193  <.0001
vwfree          1 505.0665  <.0001
FACPERM        89   0.6285  0.9975
```

```
insbetafx1$sigma^2
```

```
[1] 92.63053
```

```
AIC(insbetafx1)
```

```
[1] 39672.63
```

```
logLik(insbetafx1)*(-2)
```

```
'log Lik.' 39488.63 (df=92)
```

```
#VARIALBE SLOPES MODEL
insbetafx2<-gls(retfree~vwfree*FACPERM-vwfree-FACPERM, method="REML", data=insbeta)
anova(insbetafx2)
```

```
Denom. DF: 5309
               numDF   F-value p-value
(Intercept)        1 24.712995  <.0001
vwfree:FACPERM    90  7.562791  <.0001
```

```
insbetafx2$sigma^2
```

```
[1] 90.78022
```

```
AIC(insbetafx2)
```

```
[1] 39830.52
```

```
logLik(insbetafx2)*(-2)
```

```
'log Lik.' 39646.52 (df=92)
```

```
#VARIABLE INTERCEPTS AND SLOPES MODEL
insbetafx3<-gls(retfree~vwfree*FACPERM, method="REML", data=insbeta)
anova(insbetafx3)
```

```
Denom. DF: 5220
               numDF  F-value p-value
(Intercept)        1  24.6569  <.0001
vwfree             1 514.1906  <.0001
FACPERM           89   0.6399  0.9966
vwfree:FACPERM    89   2.0776  <.0001
```

```
insbetafx3$sigma^2
```

```
[1] 90.98683
```

```
AIC(insbetafx3)
```

```
[1] 39712.59
```

```
logLik(insbetafx3)*(-2)
```

```
'log Lik.' 39350.59 (df=181)
```

```
#VARIABLE SLOPES MODEL WITH AR(1) TERM
insbetafx4<-gls(retfree~vwfree:FACPERM, data=insbeta, method="REML", correlation=corAR
anova(insbetafx4)
```

```
Denom. DF: 5309
               numDF   F-value p-value
(Intercept)        1 29.285237  <.0001
vwfree:FACPERM    90  7.941803  <.0001
```

```
insbetafx4$sigma^2
```

```
[1] 90.76872
```

```
AIC(insbetafx4)
```

```
[1] 39796.92
```

```
logLik(insbetafx4)*(-2)
```

```
'log Lik.' 39610.92 (df=93)
```

```
insbetafx4$modelStruct
```

```
corStruct  parameters:
[1] -0.1689266
```

Table 8.4 summarizes the fit of each model. Based on these fits, we will use the variable slopes with an $AR(1)$ error term model as the baseline for investigating time-varying coefficients.

Then we can include random effects:

```
insbetarm<-lme(retfree~vwfree, data=insbeta, random=~vwfree-1|PERMNO) #Random - Effect

insbetarco<-lme(retfree~vwfree, data=insbeta, random=~1+vwfree|PERMNO, correlation=cor

#due to convergence problem, I add the "control = lmeControl(opt = "optim")".

#Random - Coefficients Model
summary(insbetarm)
```

```
Linear mixed-effects model fit by REML
 Data: insbeta
       AIC      BIC    logLik
  39738.53 39764.91 -19865.27

Random effects:
 Formula: ~vwfree - 1 | PERMNO
          vwfree Residual
StdDev: 0.2569603 9.527865
```

```
Fixed effects: retfree ~ vwfree
                 Value  Std.Error   DF   t-value p-value
(Intercept) -0.5644229 0.14016877 5309 -4.026737   1e-04
vwfree       0.7179819 0.04164033 5309 17.242464   0e+00
 Correlation:
       (Intr)
vwfree -0.289

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-7.18150077 -0.49947031 -0.02643177  0.46193572 10.17362517

Number of Observations: 5400
Number of Groups: 90
```

```r
summary(insbetarco)
```

```
Linear mixed-effects model fit by REML
 Data: insbeta
      AIC      BIC   logLik
  39697.8 39743.95 -19841.9

Random effects:
 Formula: ~1 + vwfree | PERMNO
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 0.5759112 (Intr)
vwfree      0.3182517 -0.831
Residual    9.5058076

Correlation Structure: AR(1)
 Formula: ~1 | PERMNO
 Parameter estimate(s):
        Phi
-0.08830483
Fixed effects: retfree ~ vwfree
                 Value  Std.Error   DF   t-value p-value
(Intercept) -0.5905640 0.14322023 5309 -4.123468       0
vwfree       0.7378101 0.04596025 5309 16.053222       0
 Correlation:
       (Intr)
vwfree -0.508

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
```

```
-7.20057083 -0.49733487 -0.02677384  0.46069650 10.22355808

Number of Observations: 5400
Number of Groups: 90
```

Cleaning up companies with more than one Ticker names but having the same `PERMNO`:

```
tab<-as.matrix(xtabs(~PERMNO+TICKER, insbeta)) #a logical matrix cross-tabulation of P
which(rowSums(tab>0)>1)
```

```
10085 10388 10933 11203 11371 11406 11713 22198 37226 48901 52936 58393
    1     5    10    12    13    14    16    24    30    41    44    50
60687 76099 76697 77052 77815
   56    72    79    83    86
```

```
# PERMNOs that have more than one ticker
#10085 10388 10933 11203 11371 11406 11713 22198 37226 48901 52936 58393 60687
#    1     5    10    12    13    14    16    24    30    41    44    50    56
#76099 76697 77052 77815
#   72    79    83    86
# For each PERMNO go through the following code check on the the TICKER names and freq
# which(tab["10388",]>0)
#TREN  TWK
#  96   99
#> tab["10388", c(96,99)]
# TREN  TWK
#  57    3  # THIS SHOWS THE FREQUENCY AS WELL AS THE TICKER NAMES FOR ONE SINGLE PERM
```

Recode Tickers:

```
insbeta$TICKER[insbeta$PERMNO=="10085"]<-"UICI"
insbeta$TICKER[insbeta$PERMNO=="10388"]<-"TREN"
insbeta$TICKER[insbeta$PERMNO=="10933"]<-"MKL"
insbeta$TICKER[insbeta$PERMNO=="11203"]<-"PXT"
insbeta$TICKER[insbeta$PERMNO=="11371"]<-"HCCC"
insbeta$TICKER[insbeta$PERMNO=="11406"]<-"CSH"
insbeta$TICKER[insbeta$PERMNO=="11713"]<-"PTAC"
insbeta$TICKER[insbeta$PERMNO=="22198"]<-"CRLC"
insbeta$TICKER[insbeta$PERMNO=="37226"]<-"FOM"
insbeta$TICKER[insbeta$PERMNO=="48901"]<-"MLA"
insbeta$TICKER[insbeta$PERMNO=="52936"]<-"MCY"
insbeta$TICKER[insbeta$PERMNO=="58393"]<-"RLR"
insbeta$TICKER[insbeta$PERMNO=="60687"]<-"AFG"
insbeta$TICKER[insbeta$PERMNO=="76099"]<-"DFG"
insbeta$TICKER[insbeta$PERMNO=="76697"]<-"FHS"
insbeta$TICKER[insbeta$PERMNO=="77052"]<-"UWZ"
insbeta$TICKER[insbeta$PERMNO=="77815"]<-"EQ"
```

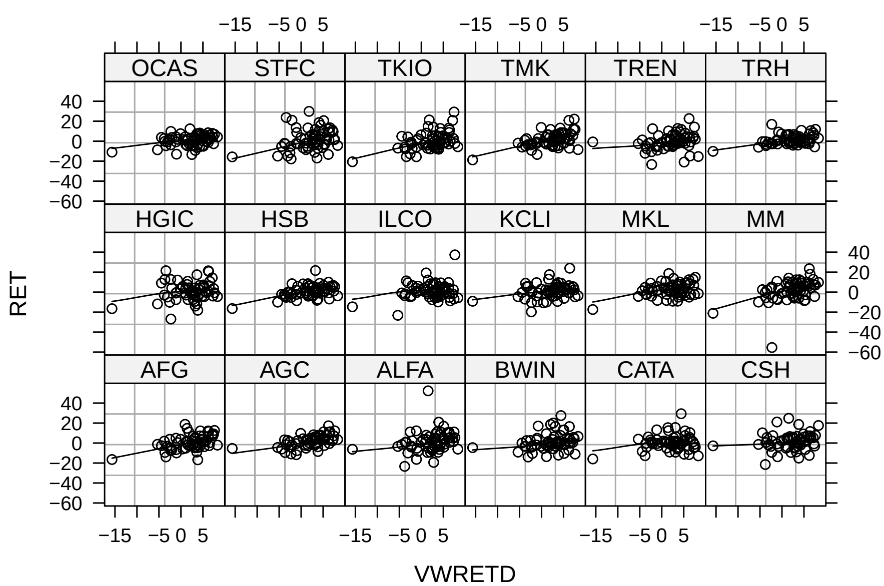
Retuen the following checking the consistency between `PERMNO` and `TICKER`:

```
tab<-as.matrix(xtabs(~PERMNO+TICKER, insbeta))
which(rowSums(tab>0)>1) #RESULT SHOULD BE ZERO
```

```
named integer(0)
```

7.2.7 Figure 8.1: Trellis plot of returns versus market return

```
#PRODUCE A TRELLIS PLOT TO SHOW VARYING BETAS
library(lattice)
insbeta$ID=factor(insbeta$PERMNO)
insbeta$TK=factor(insbeta$TICKER)
sampbeta <- subset(insbeta, ID %in% sample(levels(insbeta$ID), 18, replace=FALSE) )

xyplot(RET ~ VWRETD | TK, data=sampbeta, layout=c(6,3,1), panel = function(x, y) {
 panel.grid()
 panel.xyplot(x, y)
 panel.loess(x, y, span = 1.5)
 })
```



Chapter 8

Binary Dependent Variables

8.1 Import Data

```
taxprep=read.table("TXTData/TaxPrep.txt", sep ="\t", quote = "",header=TRUE)

#taxprep=read.table(choose.files(), header=TRUE, sep="\t")
```

Data for this study are from the Statistics of Income (SOI) Panel of Individual Returns, a part of the Ernst and Young/University of Michigan Tax Research Database. The SOI Panel represents a simple random sample of unaudited individual income tax returns filed for tax years 1979-1990. The data are compiled from a stratified probability sample of unaudited individual income tax returns, Forms 1040, 1040A and 1040EZ, filed by U.S. taxpayers. The estimates that are obtained from these data are intended to represent all returns filed for the income tax years under review. All returns processed are subjected to sampling except tentative and amended returns.

Variable	Description
MS	is an indicator variable of the taxpayer's marital status. It is coded one if the taxpayer is married and zero otherwise.
HH	is an indicator variable, one if the taxpayer is a head of household and zero otherwise.
DEPEND	is the number of dependents claimed by the taxpayer.
AGE	is the presence of an indicator for age 65 or over.
F1040A	is an indicator variable of the taxpayer's filing type. It is coded one if the taxpayer uses Form 1040A and zero otherwise.
F1040EZ	is an indicator variable of the taxpayer's filing type. It is coded one if the taxpayer uses Form 1040EZ and zero otherwise.
TPI	is the sum of all positive income line items on the return.

Variable	Description
TXRT	is a marginal tax rate. It is computed on TPI less exemptions and the standard deduction.
MR	is an exogenous marginal tax rate. It is computed on TPI less exemptions and the standard deduction.
EMP	is an indicator variable, one if Schedule C or F is present and zero otherwise. Self-employed taxpayers have greater need for professional assistance to reduce the reporting risks of doing business.
PREP	is a variable indicating the presence of a paid preparer.
TAX	is the tax liability on the return.
SUBJECT	Subject identifier, 1-258.
TIME	Time identifier, 1-5.
LNTAX	is the natural logarithm of the tax liability on the return.
LNTPI	is the natural logarithm of the sum of all positive income line items on the return.

8.2 Example: Income Tax Payments and Tax Preparers (page 326)

8.2.1 TABLE 9.2. Means for binary variables

```
library(Hmisc)
summarize(taxprep$MS, taxprep$PREP, mean)
```

```
  taxprep$PREP taxprep$MS
1            0  0.5424739
2            1  0.7092084
```

```
summarize(taxprep$HH, taxprep$PREP, mean)
```

```
  taxprep$PREP taxprep$HH
1            0 0.10581222
2            1 0.06623586
```

```
summarize(taxprep$AGE, taxprep$PREP, mean)
```

```
  taxprep$PREP taxprep$AGE
1            0  0.07153502
2            1  0.16478191
```

```
summarize(taxprep$EMP, taxprep$PREP, mean)
```

```
  taxprep$PREP taxprep$EMP
1            0   0.0923994
2            1   0.2116317
```

Table 9.2 shows that those taxpayers using a professional tax preparer (PREP = 1) were more likely to be married, not the head of a household, age 65 and over, and self-employed.

8.2.2 TABLE 9.3. Summary stats for other variables

```
library(nlme)
gsummary(taxprep[, c("DEPEND", "LNTPI", "MR")], groups=taxprep$PREP, FUN=mean)
```

```
    DEPEND    LNTPI       MR
0 2.266766  9.73151 21.98733
1 2.584814 10.05881 25.18821
```

```
gsummary(taxprep[, c("DEPEND", "LNTPI", "MR")], groups=taxprep$PREP, FUN=min)
```

```
  DEPEND       LNTPI MR
0      0 -0.12751332  0
1      0 -0.09166719  0
```

```
gsummary(taxprep[, c("DEPEND", "LNTPI", "MR")], groups=taxprep$PREP, FUN=max)
```

```
  DEPEND    LNTPI MR
0      6 12.04322 50
1      6 13.22203 50
```

```
gsummary(taxprep[, c("DEPEND", "LNTPI", "MR")], groups=taxprep$PREP, FUN=sd)
```

```
    DEPEND    LNTPI       MR
0 1.300545 1.088713 11.16809
1 1.358360 1.219911 11.53564
```

Table 9.3 shows that those taxpayers using a professional tax preparer had more dependents, had a larger income, and were in a higher tax bracket.

8.2.3 TABLE 9.4. Frequency tables for some of the binary variables

```
xtabs(~taxprep$PREP+taxprep$EMP, data=taxprep)
```

```
            taxprep$EMP
taxprep$PREP   0   1
           0 609  62
           1 488 131
```

Table 9.4 provides additional information about the relation between EMP and PREP.

8.2.4 DISPLAY 9.1 Fit the logistic distribution function using maximum likelihood

```
library(Hmisc)
library(rms)
# `rms` is an R package that is a replacement for the `Design` package.
preplogit<-lrm(PREP~LNTPI+MR+EMP, data=taxprep)
preplogit
```

```
Logistic Regression Model

 lrm(formula = PREP ~ LNTPI + MR + EMP, data = taxprep)

                      Model Likelihood     Discrimination    Rank Discrim.
                         Ratio Test           Indexes           Indexes
 Obs          1290    LR chi2      67.24    R2       0.068    C       0.642
  0            671    d.f.             3    g        0.512    Dxy     0.283
  1            619    Pr(> chi2) <0.0001    gr       1.668    gamma   0.283
 max |deriv| 2e-10                          gp       0.121    tau-a   0.141
                                            Brier    0.236

           Coef    S.E.   Wald Z Pr(>|Z|)
 Intercept -2.3447 0.7754 -3.02  0.0025
 LNTPI      0.1881 0.0940  2.00  0.0455
 MR         0.0108 0.0088  1.22  0.2212
 EMP        1.0091 0.1693  5.96  <0.0001
```

```
# ALTERNATIVE - FIT A GENERALIZED LINEAR MODEL;
prepglm<-glm(PREP~LNTPI+MR+EMP, binomial(link=logit), data=taxprep)
prepglm
```

```

Call:  glm(formula = PREP ~ LNTPI + MR + EMP, family = binomial(link = logit),
    data = taxprep)

Coefficients:
(Intercept)        LNTPI           MR          EMP
   -2.34471      0.18811      0.01081      1.00906

Degrees of Freedom: 1289 Total (i.e. Null);  1286 Residual
Null Deviance:      1786
Residual Deviance: 1719     AIC: 1727
```

Display 9.1 shows a fitted logistic regression model, using `LNTPI`, `MR`, and `EMP` as explanatory variables. The calculations were done using SAS PROC LOGISTIC.

8.3 SECTION 9.2 Random effects nonlinear mixed effects model

```r
library(glmmML)
# nlme can not be used to fit a mixed effects model with responses as binomially distributed
# In R nlme can be used to estimate a mechanistic model of the relationship between response and
# install library glmmML: menu - packages - install package(s) from CRAN - glmmML
# glmmML estimates generalized linear model with random intercepts using Maximum Likelihood
# and numerical integration via Gauss-Hermite quadrature.
prepglmml<-glmmML(PREP~LNTPI+MR+EMP, binomial(link=logit), data=taxprep, cluster=taxprep$SUBJECT)
prepglmml
```

```
Call:  glmmML(formula = PREP ~ LNTPI + MR + EMP, family = binomial(link = logit),      data = tax

                coef se(coef)       z Pr(>|z|)
(Intercept) -3.11544  1.43807 -2.1664  0.03030
LNTPI        0.22805  0.16531  1.3795  0.16800
MR           0.01394  0.02116  0.6591  0.51000
EMP          1.79380  0.56817  3.1572  0.00159

Scale parameter in mixing distribution:  4.454 gaussian
Std. Error:                              0.1963

        LR p-value for H_0: sigma = 0:  7.788e-145

Residual deviance: 1064 on 1285 degrees of freedom  AIC: 1074
```

8.3.1 Generalized linear mixed effects model

```r
# FIT GLMM with multivariate normal random effects, using Penalized Quasi-Likelihood
library(lme4)
prepGLMM<-glmer(PREP~LNTPI+MR+EMP+ (1|SUBJECT), family=binomial(link=logit), data=taxprep)
```

8.4 SECTION 9.3 Fixed effect model

```r
taxprep$facsub<-factor(taxprep$SUBJECT)
# The fixed - effects model did not converge under maximum likelihood method, because of the `fac
# prepfxlogit<-lrm(PREP~LNTPI+MR+EMP+facsub,data=taxprep)
# I assume we can use glm() to fit the model.
prepfxlogit<-glm(PREP~LNTPI+MR+EMP+facsub,family=binomial(link=logit),data=taxprep)
```

8.5 SECTION 9.4 Marginal model and generalized equation estimation

```
library(gee)
prepgee1<-gee(PREP ~ LNTPI+MR+EMP, id=SUBJECT, data=taxprep, family=binomial(link=logi
```

```
Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27

running glm to get initial regression estimate

(Intercept)       LNTPI          MR         EMP 
-2.34471453  0.18810526  0.01081409  1.00906337 
```

```
#gee Results match with SAS results
summary(prepgee1)
```

```

 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998) 

Model:
 Link:                      Logit 
 Variance to Mean Relation: Binomial 
 Correlation Structure:     Exchangeable 

Call:
gee(formula = PREP ~ LNTPI + MR + EMP, id = SUBJECT, data = taxprep, 
    family = binomial(link = logit), corstr = "exchangeable")

Summary of Residuals:
       Min         1Q     Median         3Q        Max 
-0.8131251 -0.4480400 -0.2898825  0.5079648  0.9138800 


Coefficients:
               Estimate  Naive S.E.   Naive z Robust S.E.   Robust z
(Intercept) -2.34471453 0.779479227 -3.008053  1.13139184 -2.0724160
LNTPI        0.18810526 0.094523103  1.990045  0.13685915  1.3744442
MR           0.01081409 0.008886585  1.216901  0.01122493  0.9633996
EMP          1.00906337 0.170162931  5.929983  0.17813257  5.6646764

Estimated Scale Parameter:  1.010469
Number of Iterations:  1

Working Correlation
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
```

```
[2,]    0    0    0    0    0
[3,]    0    0    0    0    0
[4,]    0    0    0    0    0
[5,]    0    0    0    0    0
```

```r
prepgee2<-gee(PREP ~ LNTPI+MR+EMP, id=SUBJECT, data=taxprep, family=binomial(link=logit), corstr=
```

```
Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
running glm to get initial regression estimate

(Intercept)       LNTPI          MR         EMP 
-2.34471453  0.18810526  0.01081409  1.00906337 
```

```r
summary(prepgee2)
```

```

 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998) 

Model:
 Link:                      Logit 
 Variance to Mean Relation: Binomial 
 Correlation Structure:     Unstructured 

Call:
gee(formula = PREP ~ LNTPI + MR + EMP, id = SUBJECT, data = taxprep, 
    family = binomial(link = logit), corstr = "unstructured")

Summary of Residuals:
       Min         1Q     Median         3Q        Max 
-0.8131251 -0.4480400 -0.2898825  0.5079648  0.9138800 


Coefficients:
               Estimate  Naive S.E.   Naive z Robust S.E.   Robust z
(Intercept) -2.34471453 0.779479227 -3.008053  1.13139184 -2.0724160
LNTPI        0.18810526 0.094523103  1.990045  0.13685915  1.3744442
MR           0.01081409 0.008886585  1.216901  0.01122493  0.9633996
EMP          1.00906337 0.170162931  5.929983  0.17813257  5.6646764

Estimated Scale Parameter:  1.010469
Number of Iterations:  1

Working Correlation
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0    0    0    0    0
```

```
[3,]    0    0    0    0    0
[4,]    0    0    0    0    0
[5,]    0    0    0    0    0
```

Chapter 9

Generalized Linear Models

9.1 Import Data

```
#tfiling=read.table("c:\\data\\tfiling.txt", header=TRUE, sep="\t") # the two missing observatio

tfiling.na=read.table("TXTData/TFiling.txt", sep ="\t", quote = "",header=TRUE)
tfiling<-na.omit(tfiling.na)
tfiling$GSTATEP=tfiling$GSTATEP/10000
tfiling$POP=tfiling$POPULATI/1000
tfiling$YEAR=tfiling$TIME+1983
```

There is a widespread belief that, in the United States, parties have become increasingly willing to go to the judicial system to settle disputes. This is particularly true in the insurance industry, an industry designed to spread risk among individuals who are subject to unfortunate events that threaten their livelihoods. Litigation in the insurance industry arises from two types of disagreement among parties, breach of faith and tort. A breach of faith is a failure by a party to the contract to perform according to its terms. This type of dispute is relatively confined to issues of facts including the nature of the duties and the action of each party. A tort action is a civil wrong, other than breach of contract, for which the court will provide a remedy in the form of action for damages. A civil wrong may include malice, wantonness oppression or capricious behavior by a party. Generally, much larger damages can be collected for tort actions because the award may be large enough to "sting" the guilty party. Since large insurance companies are viewed as having "deep pockets," these awards can be quite large indeed.

Variable	Description
FILINGS	Number of filings of tort actions against insurance companies.
POPLAWYR	The population per lawyer.
VEHCMILE	Number of automobiles miles per mile of road, in thousands.
GSTATEP	Percentage of gross state product from manufacturing and construction.
POPDENSY	Number of people per ten square miles of land.
WCMPMAX	Maximum workers' compensation weekly benefit.
URBAN	Percentage of population living in urban areas.
UNEMPLOY	State unemployment rate, in percentages.
J&SLIAB	An indicator of joint and several liability reform.
COLLRULE	An indicator of collateral source reform.
CAPS	An indicator of caps on non-economic reform.
PUNITIVE	An indicator of limits of punitive damage.
TIME	Year identifier, 1-6
STATE	State identifier, 1-19.

9.2 Example: Tort Filings (Page 356)

There is a widespread belief that, in the United States, contentious parties have become increasingly willing to go to the judicial system to settle disputes. This is particularly true when one party is from the insurance industry, an industry designed to spread risk among individuals. Litigation in the insurance industry arises from two types of disagreement among parties, breach of faith and tort. A breach of faith is a failure by a party to the contract to perform according to its terms. A tort action is a civil wrong, other than breach of contract, for which the court will provide a remedy in the form of action for damages. A civil wrong may include malice, wantonness, oppression, or capricious behavior by a party. Generally, large damages can be collected for tort actions because the award may be large enough to "sting" the guilty party. Because large insurance companies are viewed as having "deep pockets," these awards can be quite large.

9.2.1 TABLE 10.3 Averages with explanatory binary variables

```
library(Hmisc)
summary(tfiling[, c("JSLIAB", "COLLRULE", "CAPS", "PUNITIVE")])
```

```
     JSLIAB          COLLRULE           CAPS           PUNITIVE     
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000  
 Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000  
 Mean   :0.4911   Mean   :0.3036   Mean   :0.2321   Mean   :0.3214  
 3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:1.0000  
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000  
```

```
summarize(tfiling$NUMFILE, tfiling$JSLIAB, mean)
```

```
  tfiling$JSLIAB tfiling$NUMFILE
1              0        15330.07
2              1        25886.76
```

```
summarize(tfiling$NUMFILE, tfiling$COLLRULE, mean)
```

```
  tfiling$COLLRULE tfiling$NUMFILE
1                0        20726.64
2                1        20026.71
```

```
summarize(tfiling$NUMFILE, tfiling$CAPS, mean)
```

```
  tfiling$CAPS tfiling$NUMFILE
1            0       24682.488
2            1        6726.615
```

```
summarize(tfiling$NUMFILE, tfiling$PUNITIVE, mean)
```

```
  tfiling$PUNITIVE tfiling$NUMFILE
1                0        17693.38
2                1        26469.14
```

In Table 10.3 we see that 23.2% of the 112 stateyear observations were under limits (caps) on noneconomic reform. Those observations not under limits on noneconomic reforms had a larger average number of filings.

9.2.2 TABLE 10.4 Summary statistics for other variables

```
summary(tfiling[,c("NUMFILE", "POP", "POPLAWYR", "VEHCMILE", "GSTATEP", "POPDENSY", "WCMPMAX", "U
```

```
    NUMFILE           POP            POPLAWYR        VEHCMILE    
 Min.   :  512   Min.   : 0.521   Min.   :211.0   Min.   :  63.0  
 1st Qu.: 1790   1st Qu.: 1.109   1st Qu.:315.8   1st Qu.: 267.0  
```

```
 Median :  9085   Median : 3.353   Median :382.5   Median : 510.5  
 Mean   : 20514   Mean   : 6.679   Mean   :377.3   Mean   : 654.8  
 3rd Qu.: 31227   3rd Qu.:10.752   3rd Qu.:426.2   3rd Qu.: 933.5  
 Max.   :137455   Max.   :29.064   Max.   :537.0   Max.   :1899.0  
    GSTATEP          POPDENSY          WCMPMAX           URBAN       
 Min.   : 1.000   Min.   :   0.90   Min.   : 203.0   Min.   : 18.90  
 1st Qu.: 1.982   1st Qu.:  20.75   1st Qu.: 275.8   1st Qu.: 44.98  
 Median : 6.243   Median :  63.90   Median : 319.0   Median : 78.90  
 Mean   :12.667   Mean   : 168.18   Mean   : 350.0   Mean   : 69.36  
 3rd Qu.:17.673   3rd Qu.: 212.00   3rd Qu.: 382.0   3rd Qu.: 90.50  
 Max.   :69.738   Max.   :1043.00   Max.   :1140.0   Max.   :100.00  
    UNEMPLOY     
 Min.   : 2.600  
 1st Qu.: 5.075  
 Median : 5.950  
 Mean   : 6.217  
 3rd Qu.: 7.225  
 Max.   :10.800  
```

```
cor(tfiling$NUMFILE, tfiling[, c("POP", "POPLAWYR", "VEHCMILE", "GSTATEP", "POPDENSY",
```

```
          POP   POPLAWYR  VEHCMILE   GSTATEP  POPDENSY    WCMPMAX
[1,] 0.901947 -0.3781212 0.5175764 0.9145287 0.3678268 -0.2655063
         URBAN    UNEMPLOY    JSLIAB    COLLRULE       CAPS  PUNITIVE
[1,] 0.5501013 0.007600309 0.1825544 -0.01113243 -0.2622334 0.1417713
```

The correlations in Table 10.4 show that several of the economic and demographic variables appear to be related to the number of filings. In particular, we note that the number of filings is highly related to the state population.

9.3 Section 10.2 Homogeneous model

```
tfiling$POPLAWYR <- tfiling$POPLAWYR/1000
tfiling$VEHCMILE <- tfiling$VEHCMILE/1000
tfiling$GSTATEP<- tfiling$GSTATEP/1000
tfiling$POPDENSY<-tfiling$POPDENSY/1000
tfiling$WCMPMAX<-tfiling$WCMPMAX/1000
tfiling$URBAN<-tfiling$URBAN/1000
tfiling$LNPOP<-log(tfiling$POPULATI*1000)
```

9.3.1 TABLE 10.5 Tort filings model coefficient estimates

```
glm(NUMFILE ~ POPLAWYR+VEHCMILE+POPDENSY+WCMPMAX+URBAN+UNEMPLOY+JSLIAB+COLLRULE+CAPS+PU
```

```
Call:  glm(formula = NUMFILE ~ POPLAWYR + VEHCMILE + POPDENSY + WCMPMAX +
    URBAN + UNEMPLOY + JSLIAB + COLLRULE + CAPS + PUNITIVE, family = poisson(link = "log"),
    data = tfiling, offset = LNPOP)

Coefficients:
(Intercept)     POPLAWYR     VEHCMILE     POPDENSY      WCMPMAX
   -7.94343      2.16331      0.86188      0.39182     -0.80195
      URBAN     UNEMPLOY       JSLIAB     COLLRULE         CAPS
    0.89183      0.08664      0.17678     -0.02982     -0.03193
   PUNITIVE
    0.02953

Degrees of Freedom: 111 Total (i.e. Null);  101 Residual
Null Deviance:      430300
Residual Deviance: 118300   AIC: 119500
```

```
tfiling$TIMEFAC<-factor(tfiling$TIME)
glm(NUMFILE ~ TIMEFAC+POPLAWYR+VEHCMILE+POPDENSY+WCMPMAX+URBAN+UNEMPLOY+JSLIAB+COLLRULE+CAPS+PUNI
```

```
Call:  glm(formula = NUMFILE ~ TIMEFAC + POPLAWYR + VEHCMILE + POPDENSY +
    WCMPMAX + URBAN + UNEMPLOY + JSLIAB + COLLRULE + CAPS + PUNITIVE -
    1, family = poisson(link = "log"), data = tfiling, offset = LNPOP)

Coefficients:
TIMEFAC1  TIMEFAC2  TIMEFAC3  TIMEFAC4  TIMEFAC5  TIMEFAC6  POPLAWYR
-7.97398  -7.90048  -7.83975  -7.92226  -7.88501  -7.88776   2.12339
VEHCMILE  POPDENSY   WCMPMAX     URBAN  UNEMPLOY    JSLIAB  COLLRULE
 0.85617   0.38357  -0.82607   0.97667   0.08605   0.12953  -0.02347
    CAPS  PUNITIVE
-0.05575   0.05281

Degrees of Freedom: 112 Total (i.e. Null);  96 Residual
Null Deviance:      1.465e+09
Residual Deviance: 115500   AIC: 116700
```

Table 10.5 summarizes the fit of three Poisson models. With the basic homogeneous Poisson model, all explanatory variables turn out to be statistically significant, as evidenced by the small p-values. However, the Poisson model assumes that the variance equals the mean; this is often a restrictive assumption for empirical work. Thus, to account for potential overdispersion, Table 10.5 also summarizes a homogenous Poisson model with an estimated scale parameter. Table 10.5 emphasizes that, although the regression coefficient estimates do not change with the introduction of the scale parameter, estimated standard errors and thus p-values do change.

9.4 Section 10.3 Marginal Models

9.4.1 With in state correlation independent

```
library(gee)
gee(NUMFILE ~ offset(LNPOP)+POPLAWYR+VEHCMILE+POPDENSY+WCMPMAX+URBAN+UNEMPLOY+JSLIAB+CO
```

```
Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27

running glm to get initial regression estimate

(Intercept)    POPLAWYR    VEHCMILE    POPDENSY     WCMPMAX       URBAN
-7.94343077  2.16331290  0.86187552  0.39181865 -0.80195312  0.89182723
   UNEMPLOY      JSLIAB    COLLRULE        CAPS    PUNITIVE
 0.08663651  0.17677542 -0.02982377 -0.03193075  0.02952586


 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998)

Model:
 Link:                      Logarithm
 Variance to Mean Relation: Poisson
 Correlation Structure:     Independent

Call:
gee(formula = NUMFILE ~ offset(LNPOP) + POPLAWYR + VEHCMILE +
    POPDENSY + WCMPMAX + URBAN + UNEMPLOY + JSLIAB + COLLRULE +
    CAPS + PUNITIVE, id = STATE, data = tfiling, family = poisson(link = "log"),
    corstr = "independence")

Number of observations :  112

Maximum cluster size   :  6


Coefficients:
(Intercept)    POPLAWYR    VEHCMILE    POPDENSY     WCMPMAX       URBAN
-7.94343079  2.16331290  0.86187552  0.39181865 -0.80195312  0.89182735
   UNEMPLOY      JSLIAB    COLLRULE        CAPS    PUNITIVE
 0.08663651  0.17677542 -0.02982377 -0.03193075  0.02952586

Estimated Scale Parameter:  1285.7
Number of Iterations:  1

Working Correlation[1:4,1:4]
     [,1] [,2] [,3] [,4]
```

```
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1


Returned Error Value:
[1] 0
```

```
gee(NUMFILE ~ offset(LNPOP)+POPLAWYR+VEHCMILE+POPDENSY+WCMPMAX+URBAN+UNEMPLOY+JSLIAB+COLLRULE+CAP
```

```
Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
running glm to get initial regression estimate

(Intercept)    POPLAWYR    VEHCMILE    POPDENSY     WCMPMAX       URBAN
-7.94343077  2.16331290  0.86187552  0.39181865 -0.80195312  0.89182723
   UNEMPLOY      JSLIAB    COLLRULE        CAPS    PUNITIVE
 0.08663651  0.17677542 -0.02982377 -0.03193075  0.02952586


 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998)

Model:
 Link:                      Logarithm
 Variance to Mean Relation: Poisson
 Correlation Structure:     AR-M , M = 1

Call:
gee(formula = NUMFILE ~ offset(LNPOP) + POPLAWYR + VEHCMILE +
    POPDENSY + WCMPMAX + URBAN + UNEMPLOY + JSLIAB + COLLRULE +
    CAPS + PUNITIVE, id = STATE, data = tfiling, family = poisson(link = "log"),
    corstr = "AR-M", Mv = 1)

Number of observations :  112

Maximum cluster size   :  6


Coefficients:
(Intercept)    POPLAWYR    VEHCMILE    POPDENSY     WCMPMAX       URBAN
-7.99997854  1.88219159  0.69338537  0.37164593  0.05604892  4.93610043
   UNEMPLOY      JSLIAB    COLLRULE        CAPS    PUNITIVE
 0.04340498  0.17025340 -0.06500658  0.09194548 -0.04663443

Estimated Scale Parameter:  1444.921
Number of Iterations:  9
```

```
Working Correlation[1:4,1:4]
          [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.8517403 0.7254616 0.6179048
[2,] 0.8517403 1.0000000 0.8517403 0.7254616
[3,] 0.7254616 0.8517403 1.0000000 0.8517403
[4,] 0.6179048 0.7254616 0.8517403 1.0000000


Returned Error Value:
[1] 0
```

```
#THE NUMBER WAS A LITTLE OFF COMPARED WITH SAS ESTIMATE
```

9.4.2 Random effects model

```
# MODEL WITHOUR RANDOM EFFECTS
glm(NUMFILE ~ POPLAWYR+VEHCMILE+POPDENSY+WCMPMAX+URBAN+UNEMPLOY+JSLIAB+COLLRULE+CAPS+PU
```

```
Call:  glm(formula = NUMFILE ~ POPLAWYR + VEHCMILE + POPDENSY + WCMPMAX +
    URBAN + UNEMPLOY + JSLIAB + COLLRULE + CAPS + PUNITIVE, family = poisson(link = "lo
    data = tfiling, offset = LNPOP)

Coefficients:
(Intercept)     POPLAWYR     VEHCMILE     POPDENSY      WCMPMAX
   -7.94343      2.16331      0.86188      0.39182     -0.80195
      URBAN     UNEMPLOY       JSLIAB     COLLRULE         CAPS
    0.89183      0.08664      0.17678     -0.02982     -0.03193
   PUNITIVE
    0.02953

Degrees of Freedom: 111 Total (i.e. Null);  101 Residual
Null Deviance:      430300
Residual Deviance: 118300   AIC: 119500
```

Chapter 10

Categorical Dependent Variables and Survival Models

10.1 Import Data

```
#yogurtbasic<-read.table(choose.files(), header=TRUE, sep="\t")

#library(Ecdat)# You need to install package 'Ecdat' for the data 'Yogurt'.
#data(Yogurt) #the data used in this Chapter.
#yogurtdata<-Yogurt
#now we need to modify the dataset
colnames(yogurtdata) = c("id","fy","fd","fh","fw","py","pd","ph","pw","choice")

yogurtdata$yoplait<-(yogurtdata$choice=="yoplait")
yogurtdata$dannon<-(yogurtdata$choice=="dannon")
yogurtdata$hiland<-(yogurtdata$choice=="hiland")
yogurtdata$weight<-(yogurtdata$choice=="weight")
```

10.2 Chap11Yogurt2013.R

```
yogurtdata<-read.csv("TXTData/yogurt.dat", header=F, sep=" ")
colnames(yogurtdata) = c("id","yoplait","dannon","weight","hiland","fy","fd","fw","fh","py","pd",
```

10.3 Table 11.2 Number of Choices

```
yogurtdata$occasion<-seq(yogurtdata$id)

yogurtdata$TYPE<-1*yogurtdata$yoplait+2*yogurtdata$dannon+3*yogurtdata$weight+4*yogurtd

yogurtdata$PRICE<-yogurtdata$py*yogurtdata$yoplait + yogurtdata$pd*yogurtdata$dannon +

yogurtdata$FEATURE<-yogurtdata$fy*yogurtdata$yoplait + yogurtdata$fd*yogurtdata$dannon

table(yogurtdata$TYPE)
```

```
  1   2   3   4 
818 970 553  71 
```

```
summary(yogurtdata[, c("fy", "fd", "fw", "fh")])[4,]
```

```
                 fy                    fd                    fw 
"Mean   :0.05597  " "Mean   :0.03773  " "Mean   :0.03773  " 
                 fh 
 "Mean   :0.0369  " 
```

Table 11.2 shows that Yoplait was the most frequently selected (33.9%) type ofyogurt in our sample whereas Hiland was the least frequently selected (2.9%). Yoplait was also the most heavily advertised, appearing in newspaper advertisements 5.6% of the time that the brand was chosen.

10.3.1 Table 11.2 Basic summary statistics for prices

```
t(summary(yogurtdata[, c("py", "pd", "pw", "ph")]))
```

```
      py Min.   :0.0030    1st Qu.:0.1030    Median :0.1080  
      pd Min.   :0.01900   1st Qu.:0.08100   Median :0.08600  
      pw Min.   :0.00400   1st Qu.:0.07900   Median :0.07900  
      ph Min.   :0.02500   1st Qu.:0.05000   Median :0.05400  
                                                             
      py Mean   :0.1068    3rd Qu.:0.1150    Max.   :0.1930  
      pd Mean   :0.08163   3rd Qu.:0.08600   Max.   :0.11100  
      pw Mean   :0.07949   3rd Qu.:0.08600   Max.   :0.10400  
      ph Mean   :0.05363   3rd Qu.:0.06100   Max.   :0.08600  
```

```
sd(as.matrix(yogurtdata[, c("py")]))
```

```
[1] 0.01906265
```

```
sd(as.matrix(yogurtdata[, c("pd")]))
```

```
[1] 0.01062886
```

```
sd(as.matrix(yogurtdata[, c("pw")]))
```

```
[1] 0.007735004
```

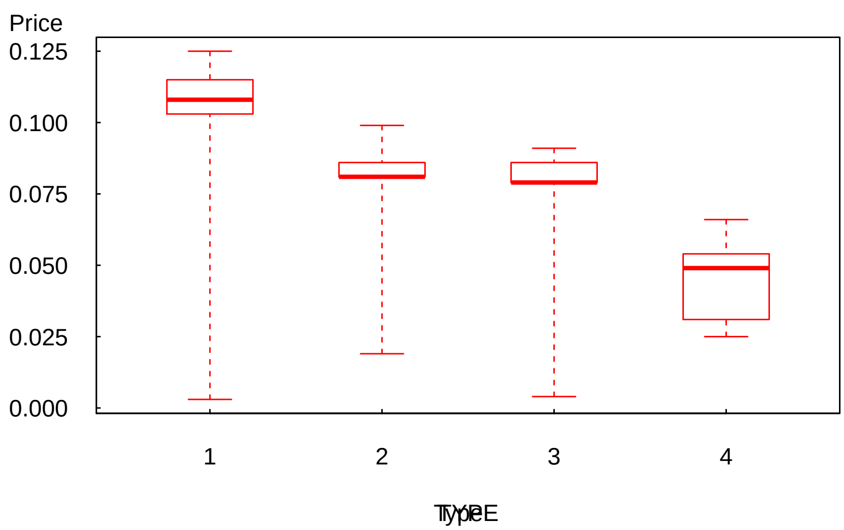
```
sd(as.matrix(yogurtdata[, c("ph")]))
```

```
[1] 0.00805391
```

Table 11.3 shows that Yoplait was also the most expensive, costing 10.7 cents per ounce, on average. Table 11.3 also shows that there are several prices that were far below the average, suggesting some potential influential observations.

10.3.2 vissualize the data

```
boxplot(PRICE~TYPE, range=0, data=yogurtdata, boxwex=0.5, border="red", yaxt="n", xaxt="n", ylab=
axis(2, at=seq(0,0.125, by=0.025), las=1, font=10, cex=0.005, tck=0.01)
axis(1, at=seq(1,4, by=1), font=10, cex=0.005, tck=0.01)
mtext("Price", side=2, adj=-1, line=5, at=0.135, font=10, las=1)
mtext("Type", side=1, adj=0, line=3, at=2.3, font=10)
box()
```

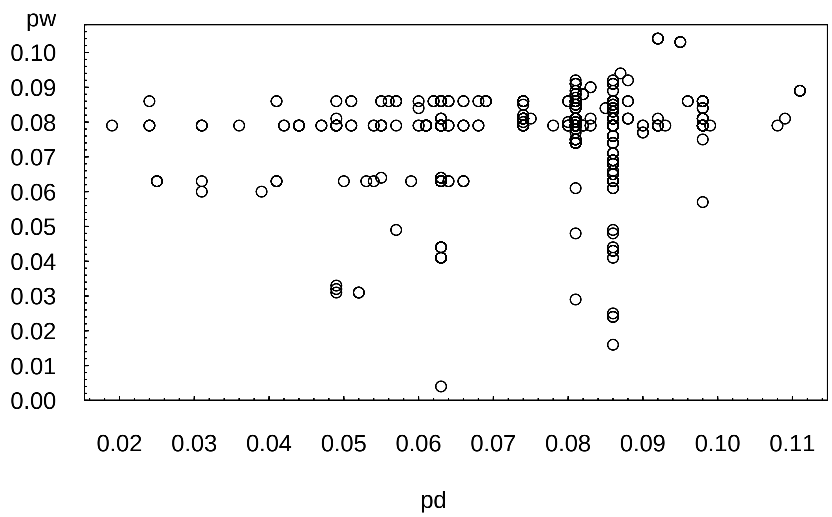


10.3.3 Note the small relationships among prices

```
cor(yogurtdata[, c("py", "pd", "pw", "ph")])
```

```
            py          pd         pw          ph
py  1.00000000  0.03201738  0.1538099 -0.01844819
pd  0.03201738  1.00000000  0.2428201 -0.04349290
pw  0.15380986  0.24282008  1.0000000 -0.02755800
ph -0.01844819 -0.04349290 -0.0275580  1.00000000
```

```
plot(pw~pd, data=yogurtdata, yaxt="n", xaxt="n", ylab="", xlab="")
axis(2, at=seq(0.00, 0.20, by=0.01), las=1, font=10, cex=0.005, tck=0.01)
axis(2, at=seq(0.00, 0.20, by=0.002),lab=F, tck=0.005)
axis(1, at=seq(0.01, 0.12, by=0.01), font=10, cex=0.005, tck=0.01)
axis(1, at=seq(0.01, 0.12, by=0.002), lab=F, tck=0.005)
mtext("pw", side=2, line=1, at=0.11, las=1, font=10)
mtext("pd", side=1, line=3, at=0.062, font=10)
```



10.3.4 More on prices

```
summary(yogurtdata$PRICE)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00300 0.07900 0.08300 0.08495 0.10300 0.12500
```

```r
range(yogurtdata$PRICE)
```

```
[1] 0.003 0.125
```

```r
which(yogurtdata$PRICE == min(yogurtdata$PRICE))
```

```
[1] 1210 1215 1930 1931 2381
```

```r
which(yogurtdata$PRICE == max(yogurtdata$PRICE))
```

```
[1]   71  952  961 1212 1213 1214 1929 2199
```

```r
library(nnet)
test <- multinom(TYPE ~ FEATURE+PRICE, data = yogurtdata)
```

```
# weights:  16 (9 variable)
initial  value 3343.741999
iter  10 value 2587.908201
iter  20 value 2364.679552
iter  30 value 2360.691887
final  value 2360.191855
converged
```

```r
summary(test)
```

```
Call:
multinom(formula = TYPE ~ FEATURE + PRICE, data = yogurtdata)

Coefficients:
  (Intercept)    FEATURE      PRICE
2    7.458657 -1.8039258  -80.44352
3    6.883787 -1.7072398  -80.32860
4    8.529886 -0.9595805 -139.53475

Std. Errors:
  (Intercept)   FEATURE    PRICE
2   0.3509758 0.2400558 3.836776
3   0.3756930 0.2673651 4.169266
4   0.4773418 0.3787246 6.632942

Residual Deviance: 4720.384
AIC: 4738.384
```

10.4 Fitting fixed effects multinomial logit model by the poisson log-linear model

```
# RESHAPE yogurtdata FROM WIDE FORMAT INTO LONG FORMAT
yogurt<-reshape(yogurtdata, varying=list(c("yoplait","dannon","weight","hiland")), v.na
"choice", idvar="occasion",timevar="brand", direction="long")
yogurt<-yogurt[order(yogurt$occasion),]
yogurt[1:8,]
```

```
    id fy fd fw fh    py    pd    pw    ph occasion TYPE PRICE FEATURE
1.1  1  0  0  0  0 0.108 0.081 0.079 0.061        1    3 0.079       0
1.2  1  0  0  0  0 0.108 0.081 0.079 0.061        1    3 0.079       0
1.3  1  0  0  0  0 0.108 0.081 0.079 0.061        1    3 0.079       0
1.4  1  0  0  0  0 0.108 0.081 0.079 0.061        1    3 0.079       0
2.1  1  0  0  0  0 0.108 0.098 0.075 0.064        2    2 0.098       0
2.2  1  0  0  0  0 0.108 0.098 0.075 0.064        2    2 0.098       0
2.3  1  0  0  0  0 0.108 0.098 0.075 0.064        2    2 0.098       0
2.4  1  0  0  0  0 0.108 0.098 0.075 0.064        2    2 0.098       0
    brand choice
1.1     1      0
1.2     2      0
1.3     3      1
1.4     4      0
2.1     1      0
2.2     2      1
2.3     3      0
2.4     4      0
```

```
yogurt$brand<-factor(yogurt$brand)
yogurt$occasion<-factor(yogurt$occasion)
# yogurtloglinear<-glm(choice~brand+occasion+FEATURE+PRICE-1, data=yogurt, family=# po
# THE ABOVE GLM INCLUDES THE FIXED EFFECTS OF THE 2412 OCCASIONS, WHICH ARE
# NUISANCE PARAMETERS, THE ESTIMATES ARE NOT OBTAINED SIMPLY BECAUSE THE
# LARGE NUMBER.
# GLM USE ITERATIVELY REWEIGHTED LEAST SQUARES TO ESTIMATE, COMPARED WITH
# GENMOD IN SAS # USING MAXIMUMLIKELIHOOD.
# DROP occasion THE GLM IS ESTIMATABLE
model1 <- glm(choice~brand+FEATURE+PRICE-1, data=yogurt, family=poisson(link="log"))
summary(model1)
```

```
Call:
glm(formula = choice ~ brand + FEATURE + PRICE - 1, family = poisson(link = "log"),
    data = yogurt)

Deviance Residuals:
```

```
     Min        1Q    Median        3Q       Max
-0.89683  -0.82357  -0.67716   0.01585   2.26052

Coefficients:
          Estimate Std. Error z value Pr(>|z|)
brand1  -1.081e+00  9.183e-02  -11.78   <2e-16 ***
brand2  -9.109e-01  9.079e-02  -10.03   <2e-16 ***
brand3  -1.473e+00  9.497e-02  -15.51   <2e-16 ***
brand4  -3.526e+00  1.459e-01  -24.16   <2e-16 ***
FEATURE  3.206e-16  8.028e-02    0.00        1
PRICE   -1.375e-13  9.840e-01    0.00        1
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 14472.0  on 9648  degrees of freedom
Residual deviance:  5665.9  on 9642  degrees of freedom
AIC: 10502

Number of Fisher Scoring iterations: 6
```

10.5 Fitting multinomial logit model with random intercepts by the possion-log-linear with random intercepts

```r
library(MASS)
# glmmPQL(choice~feature+price+occasion, data=yogurt, family=poisson(link="log"), random=~1|brand
# THE ABOVE HAS SIMILAR PROBLEM WHEN INCLUDING occasion AS FIXED EFFECTS
# OTHERWISE IT IS ESTIMATABLE IN R; HOWEVER THE RESULT IS QUITE DIFFERENT FROM # THAT OF SAS
glmmPQL(choice~FEATURE+PRICE, data=yogurt, family=poisson(link="log"), random=~1|brand)
```

```
iteration 1

iteration 2

iteration 3

iteration 4

iteration 5

iteration 6

Linear mixed-effects model fit by maximum likelihood
  Data: yogurt
  Log-likelihood: NA
```

```
  Fixed: choice ~ FEATURE + PRICE
  (Intercept)       FEATURE         PRICE
-1.743787e+00  2.005532e-14  2.982266e-13

Random effects:
 Formula: ~1 | brand
        (Intercept)  Residual
StdDev:    1.040625 0.8639861

Variance function:
 Structure: fixed weights
 Formula: ~invwt
Number of Observations: 9648
Number of Groups: 4
```