R Journal Entry Analysis

HHH

2024 2 20

# Step 1. 테스트 개요 및 목적

재무제표에 대한 회계감사의 일환으로 감사대상기간 동안 발생한 모든 전표 데이터에 대한 무결성 및 비경상적인 거래가 존재하는지를 검증

1. 계정명(FSLI) : 전체 계정
2. 기준일 (Coverage date) : 2017-01-01 ~ 2017-12-31
3. 테스트되는 경영자의 주장 (Assertion) : 완전성 (C),정확성 (A),기간귀속구분 (CO),실재성 (E/O),권리 (R),공시(PD),평가(V)

# Step 2. Test 대상 모집단

1. 전체 전표를 포함하는 분개장.
2. 모집단의 완전성 확인 : Step 5. Test 결과의 A01, A02, A03 참조

# Step 3. 오류의 정의

재무보고 프로세스 관련, 부적절하거나 비경상적인 분개 및 수정사항

# Step 4. Test 방법

* A01 . Data Integrity 검증 - 데이터 유효성을 검증하고, record에 대한 이해를 위한 절차

전표 데이터의 회계기간이 당해년도에 포함되는지 여부 검토  
전표 주요 필드값의 누락 여부 검토를 통한 data integrity 검토

* A02. 전표번호 별 차대변 일치검증  
  전표번호 별 차변금액과 대변금액이 일치하는지 확인하여 전표 데이터의 완전성을 검토  
  차변금액과 대변금액이 일치하지 않을 경우 해당 전표를 추출하여 회사 측과 확인
* A03. 시산표 Reconciliation 검증(Trial Balance Rollforward Test)  
  기초 F/S잔액에 수령한 모든 전표의 계정과목 별 합계금액을 반영하여 도출한 기말 F/S 잔액과 회사 제시 F/S와의 일치 여부 검토
* B01. 매출의 상대계정분석(매출과 연관성이 낮은 계정이 포함된 비정상적인 거래)  
  매출에 대한 상대계정분석 결과 비정상적으로 처리된 회계처리가 있는지 검토하고 해당 전표 중 특정 금액 이상인 건을 추출하여 검토

# Step 5. Test 결과

## readxl, lubridate, skimr, tidyverse 패키지를 불러온다.

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.4 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

## load Journal Entry data

* Journal Entry data 객체 load 하고 확인

je\_raw <- read\_csv('je\_utf\_colname.txt')

## Rows: 309043 Columns: 22  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (9): 계정과목명, 거래처[코드], 결의사원[코드], 유형번호[유형명], 증빙구분[구분명], INSERT\_ID, 입력일자, M...  
## dbl (11): 전표일자, 구 분[기표][번호], 전표번호, LN\_SQ, DRCR\_FG, 계정코드, 구 분[결의][년/월/일], 구 분...  
## num (2): 차변금액, 대변금액  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

je\_raw

## # A tibble: 309,043 × 22  
## 전표일자 `구 분[기표][번호]` 전표번호 LN\_SQ DRCR\_FG 계정코드 계정과목명   
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 20170101 1 201701011 1 3 12002 매출채권미수…  
## 2 20170101 1 201701011 2 4 10800 외상매출금   
## 3 20170101 1 201701011 3 4 10800 외상매출금   
## 4 20170101 46 2017010146 1 3 10800 외상매출금   
## 5 20170101 46 2017010146 2 4 40401 제품매출   
## 6 20170101 46 2017010146 3 4 25500 부가세예수금   
## 7 20170101 47 2017010147 1 3 10800 외상매출금   
## 8 20170101 47 2017010147 2 4 40401 제품매출   
## 9 20170101 47 2017010147 3 4 25500 부가세예수금   
## 10 20170101 48 2017010148 1 3 10800 외상매출금   
## # ℹ 309,033 more rows  
## # ℹ 15 more variables: 차변금액 <dbl>, 대변금액 <dbl>, `거래처[코드]` <chr>,  
## # `구 분[결의][년/월/일]` <dbl>, `구 분[결의][번호]` <dbl>,  
## # `결의부서[코드]` <dbl>, `결의사원[코드]` <chr>, `PJT[코드]` <dbl>,  
## # `유형번호[번호]` <dbl>, `유형번호[유형명]` <chr>, `증빙구분[구분명]` <chr>,  
## # INSERT\_ID <chr>, 입력일자 <chr>, MODIFY\_ID <chr>, MODIFY\_DT <chr>

### 프로그램에 맞추어 열 이름 변경

* change column name

je\_tbl <- rename(je\_raw, JEDATE = 전표일자,  
 JENO = 전표번호,  
 DR = 차변금액,  
 CR = 대변금액,  
 ACCTCD = 계정코드,  
 ACCT\_NM = 계정과목명)

### A01 Data Integrity Test

* Data skim 확인

je\_tbl <- je\_tbl |> mutate(JEDATE = ymd(JEDATE))  
skim(je\_tbl)

Data summary

|  |  |
| --- | --- |
| Name | je\_tbl |
| Number of rows | 309043 |
| Number of columns | 22 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 9 |
| Date | 1 |
| numeric | 12 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ACCT\_NM | 0 | 1.00 | 2 | 16 | 0 | 211 | 0 |
| 거래처[코드] | 2172 | 0.99 | 5 | 5 | 0 | 5909 | 0 |
| 결의사원[코드] | 0 | 1.00 | 10 | 10 | 0 | 18 | 0 |
| 유형번호[유형명] | 0 | 1.00 | 4 | 4 | 0 | 5 | 0 |
| 증빙구분[구분명] | 222980 | 0.28 | 2 | 12 | 0 | 9 | 0 |
| INSERT\_ID | 178226 | 0.42 | 10 | 10 | 0 | 17 | 0 |
| 입력일자 | 178226 | 0.42 | 21 | 22 | 0 | 20885 | 0 |
| MODIFY\_ID | 772 | 1.00 | 10 | 10 | 0 | 10 | 0 |
| MODIFY\_DT | 772 | 1.00 | 21 | 22 | 0 | 13253 | 0 |

**Variable type: Date**

| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| --- | --- | --- | --- | --- | --- | --- |
| JEDATE | 0 | 1 | 2017-01-01 | 2017-12-31 | 2017-07-10 | 326 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구 분[기표][번호] | 0 | 1.00 | 746.84 | 896.88 | 1 | 57 | 321 | 1135 | 3236 | ▇▂▁▁▁ |
| JENO | 0 | 1.00 | 65207179028.64 | 85984640164.26 | 201701011 | 2017092610 | 20170616160 | 201702221968 | 201712311321 | ▇▁▁▁▃ |
| LN\_SQ | 0 | 1.00 | 67.61 | 190.65 | 1 | 1 | 2 | 4 | 997 | ▇▁▁▁▁ |
| DRCR\_FG | 0 | 1.00 | 3.67 | 0.48 | 1 | 3 | 4 | 4 | 4 | ▁▁▁▃▇ |
| ACCTCD | 0 | 1.00 | 24348.93 | 15524.77 | 10100 | 10800 | 25301 | 40401 | 99800 | ▇▂▁▁▁ |
| DR | 207399 | 0.33 | 2856138.70 | 53914261.42 | -1000000000 | 55000 | 77000 | 200000 | 4483053872 | ▇▁▁▁▁ |
| CR | 101644 | 0.67 | 1399762.59 | 36469703.90 | -1777760000 | 9000 | 55000 | 99000 | 4483053872 | ▁▇▁▁▁ |
| 구 분[결의][년/월/일] | 0 | 1.00 | 20170680.32 | 345.45 | 20170101 | 20170406 | 20170710 | 20171011 | 20171231 | ▇▆▆▆▇ |
| 구 분[결의][번호] | 0 | 1.00 | 740.23 | 896.64 | 1 | 50 | 314 | 1111 | 3225 | ▇▂▁▁▁ |
| 결의부서[코드] | 0 | 1.00 | 1936.41 | 844.57 | 1400 | 1400 | 1400 | 2900 | 4120 | ▇▁▁▂▁ |
| PJT[코드] | 78366 | 0.75 | 3150.05 | 542.66 | 1100 | 3100 | 3100 | 3200 | 6100 | ▁▇▃▁▁ |
| 유형번호[번호] | 0 | 1.00 | 2.92 | 0.85 | 1 | 3 | 3 | 3 | 8 | ▃▇▃▁▁ |

## A02 전표번호 별 차대변 일치검증

* 전표번호, 차변금액, 대변금액 열을 선택(select)한다. 선택한 세 개의 열에 Null 값 (NA)이 있으면 0으로 모두 바꾼다.(mutate\_all)  
  전표번호별로 묶어서(group\_by) 전표번호별 차변의 합계를 DR\_sum의 열 이름으로 전표번호별 대변의 합계를 CR\_sum의 열 이름으로 정리한다.(summarise)  
  전표번호별 차변 금액과 대변 금액 합계를 각각 차이를 계산하여 Differ라는 열 이름으로 추가(mutate)한다.
* 전표번호별 차변 금액 합계와 대변 금액 합계의 차이 Differ열의 전체 합계를 계산한다.
* 계산한 차이가 0이 아닌 행을 나타낸다.

A02 <- je\_tbl |>   
 select(JENO, DR, CR) |>   
 mutate\_all(~replace(., is.na(.), 0)) |>   
 group\_by(JENO) |>   
 summarise(DR\_sum=sum(DR),  
 CR\_sum=sum(CR)) |>   
 mutate(Differ= DR\_sum - CR\_sum)  
# print(A02)  
print(A02)

## # A tibble: 80,612 × 4  
## JENO DR\_sum CR\_sum Differ  
## <dbl> <dbl> <dbl> <dbl>  
## 1 201701011 132000 132000 0  
## 2 201701021 334400 334400 0  
## 3 201701022 708400 708400 0  
## 4 201701023 31900 31900 0  
## 5 201701024 145500 145500 0  
## 6 201701025 786500 786500 0  
## 7 201701026 33000 33000 0  
## 8 201701027 13838 13838 0  
## 9 201701028 148500 148500 0  
## 10 201701029 571000 571000 0  
## # ℹ 80,602 more rows

A02 |> filter(Differ != 0)

## # A tibble: 0 × 4  
## # ℹ 4 variables: JENO <dbl>, DR\_sum <dbl>, CR\_sum <dbl>, Differ <dbl>

## A03 시산표 Reconciliation 검증(Trial Balance Rollforward Test)

### 시산표 data 준비.

* 당기 시산표 불러오기
* 전기 시산표 불러오기
* 당기시산표 columns별 결측값 확인
* 전기시산표 columns별 결측값 확인

cytb\_raw <- read\_excel('CYTB.xlsx')  
pytb\_raw <- read\_excel('PYTB.xlsx')  
colSums(is.na(cytb\_raw))

## 차 변[잔 액] DRSUM 차 변[당월계] ACCTCD   
## 0 0 0 28   
## 계정과목 대 변[당월계] CRSUM 대 변[잔 액]   
## 0 0 0 0

colSums(is.na(pytb\_raw))

## 차 변[잔 액] DRSUM 차 변[당월계] ACCTCD   
## 0 0 0 28   
## 계정과목 대 변[당월계] CRSUM 대 변[잔 액]   
## 0 0 0 0

* 결측값(Null, NA)이 ACCTCD 열에 있었으며 결측값 때문에 join에 어려움이 발생할 수 있으므로 결측을 없애고 시작한다.
* 전기 시산표에서도 결측을 없앤다.
* 결측을 없애고 나서 View(CYTB) 기능으로 읽어들인 시산표를 보니 99행이 시산표 중 재무상태표의 마지막 행이다. 시산표의 재무상태표 부분을 나누어 새로운 객체를 생성하기 위하여 1행 부터 99행까지를 썰어낸다.(slice)
* 시산표의 손익계산서 부분을 뜯어내어 새로운 객체를 생성하기 위하여 100행부터 끝까지 썰어낸다.
* 시산표에서 썰어낸(slice) 당기재무상태표 하단을 확인한다. 재무상태표 끝단이 썰렸는지 확인.
* 시산표에서 썰어낸(slice) 당기손익계산서 상단을 확인한다. 손익계산서 시작점부터 썰렸는지 확인.
* 전기시산표에서 썰어낸 전기재무상태표 하단을 확인한다. 재무상태표 끝단이 썰렸는지 확인.
* 전기 시산표의 재무상태표 부분 slice
* 당기시산표에 전기 시산표의 재무상태표 부분을 계정과목별로 맞춰서 붙인다(join). 이 때 당기시산표에 있는 계정과 전기시산표에 있는 계정이 모두 포함되도록 붙여야 한다. 전기에 변동이 있었으나 당기에 변동이 없었던 항목이 누락되지 않아야 하기 때문이다.
* 시산표의 손익계산서 부분은 당기에 발생한 잔액만 있으면 된다. 재무상태표처럼 기초잔액이라는 개념이 없다.
* 재무상태표의 잔액에서 기초잔액을 제거한 결과와 시산표의 손익계산서 부분의 잔액을 합하여 전표에서 당기에 변동이 있었던 금액을 하나로 붙인다.

cytb\_tbl <- drop\_na(cytb\_raw, ACCTCD)  
pytb\_tbl <- drop\_na(pytb\_raw, ACCTCD)  
   
CYTB\_FP <- cytb\_tbl |>   
 slice(1:99)  
CYTB\_PL <- cytb\_tbl |>   
 slice(100:n())  
PYTB\_FP <- pytb\_tbl |>   
 slice(1:103)  
   
tail(CYTB\_FP)

## # A tibble: 6 × 8  
## `차 변[잔 액]` DRSUM `차 변[당월계]` ACCTCD 계정과목 `대 변[당월계]`  
## <dbl> <dbl> <dbl> <chr> <chr> <dbl>  
## 1 0 0 0 38300 자 기… 0  
## 2 56201942 3.09e7 72053772 38900 매도가… 15851830  
## 3 0 0 0 35100 이 익 … 0  
## 4 0 0 0 37200 부의지… 0  
## 5 0 3.12e7 31216627 37400 보험수… 36377018  
## 6 0 1.32e9 0 37500 전기이… 0  
## # ℹ 2 more variables: CRSUM <dbl>, `대 변[잔 액]` <dbl>

head(CYTB\_PL)

## # A tibble: 6 × 8  
## `차 변[잔 액]` DRSUM `차 변[당월계]` ACCTCD 계정과목 `대 변[당월계]`  
## <dbl> <dbl> <dbl> <chr> <chr> <dbl>  
## 1 0 0 0 40100 상 품… 468556458  
## 2 0 0 0 40401 제 품… 3765154556  
## 3 0 0 0 40700 교 육… 671429789  
## 4 0 0 0 41100 임 대… 70591818  
## 5 0 0 0 41200 기 타… 90910  
## 6 2790687018 2.79e9 626758061 45100 상 품 … 0  
## # ℹ 2 more variables: CRSUM <dbl>, `대 변[잔 액]` <dbl>

tail(PYTB\_FP)

## # A tibble: 6 × 8  
## `차 변[잔 액]` DRSUM `차 변[당월계]` ACCTCD 계정과목 `대 변[당월계]`  
## <dbl> <dbl> <dbl> <chr> <chr> <dbl>  
## 1 0 -7.14e6 0 38800 매도가… 0  
## 2 145999555 1.87e8 187178917 38900 매도가… 41179362  
## 3 0 0 0 35100 이 익 … 0  
## 4 0 0 0 37200 부의지… 0  
## 5 0 1.50e7 15037512 37400 보험수… 3308252  
## 6 0 0 0 37500 전기이… 0  
## # ℹ 2 more variables: CRSUM <dbl>, `대 변[잔 액]` <dbl>

CYTB\_FP <- full\_join(CYTB\_FP, PYTB\_FP, by='ACCTCD') |>   
 mutate\_all(~replace(., is.na(.), 0)) |>   
 mutate(move = (DRSUM.x - CRSUM.x) -(DRSUM.y - CRSUM.y)) |>   
 select(ACCTCD, move)  
  
CYTB\_PL <- CYTB\_PL |>   
 mutate(move = (DRSUM - CRSUM)) |>   
 select(ACCTCD, move)  
  
CYTB\_move <- bind\_rows(CYTB\_FP, CYTB\_PL)

* 전표에서 계정과목별 차변금액과 대변금액 합계를 계산하여 준비한다.
* 시산표에서 준비한 당기 변동금액(전표에서 계산한 계정과목별 당기 변동 금액)을 전표에서 계산한 차변금액과 대변 금액의 합계에 붙인다.
* 차이가 0보다 큰 항목 갯수 확인.
* A03 객체에서 차이가 0이 아닌 항목을 찾아 나타낸다. 이익잉여금 항목 하나만 나타나면 이 외의 계정은 모두 일치한다.

A03 <- je\_tbl |>   
 select(ACCTCD, DR, CR) |>   
 mutate\_all(~replace(., is.na(.), 0)) |>   
 group\_by(ACCTCD) |>   
 summarise(DR\_sum=sum(DR),  
 CR\_sum=sum(CR)) |>   
 ungroup() |>  
 # join 할때 자료형 일치  
 mutate(ACCTCD = as.character(ACCTCD))  
  
A03 <- left\_join(A03, CYTB\_move, by = 'ACCTCD')  
A03 <- A03 |> mutate\_all(~replace(.,is.na(.), 0)) |>   
 mutate(Differ = (DR\_sum - CR\_sum - move))  
  
ac\_name <- je\_tbl |>   
 distinct(ACCTCD, ACCT\_NM) |>   
 mutate(ACCTCD=as.character(ACCTCD))  
  
A03 <- left\_join(A03, ac\_name, by='ACCTCD')  
  
print(A03)

## # A tibble: 211 × 6  
## ACCTCD DR\_sum CR\_sum move Differ ACCT\_NM   
## <chr> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 10100 125649815 124027515 1622300 0 현금   
## 2 10200 1735072699 1735072699 0 0 당좌예금   
## 3 10301 90863358102 90172042718 691315384 0 보통예금   
## 4 10302 785906508 439897964 346008544 0 외화예금   
## 5 10501 7800000000 8400000000 -600000000 0 정기예금   
## 6 10502 125175743 347606170 -222430427 0 정기적금   
## 7 10600 -54781102 0 -54781102 0 단기금융상품   
## 8 10800 29594408190 29451728317 142679873 0 외상매출금   
## 9 10900 101829493 -230485439 332314932 0 대손충당금(외)  
## 10 11000 45660000 41745000 3915000 0 받을어음   
## # ℹ 201 more rows

A03 |> filter(Differ != 0)

## # A tibble: 1 × 6  
## ACCTCD DR\_sum CR\_sum move Differ ACCT\_NM   
## <chr> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 37500 1320667762 0 -2957850338 4278518100 전기이월이익잉여금

### Lead Schedule xlsx 생성

amt\_sum <- je\_tbl |>   
 mutate\_all(~replace(., is.na(.), 0)) |>   
 summarise(  
 DR\_sum = sum(DR),  
 CR\_sum = sum(CR),  
 .by = ACCT\_NM  
 )   
  
# amt\_sum |> write\_xlsx("lead\_s.xlsx")

## B01 매출의 상대계정분석

* 상대계정분석에서 기준이 되는 본계정코드를 설정한다.
* 전표 데이터에서 본계정이 포함된 전표번호를 필터링(filter)한다. (행 선택)
* 전표 데이터에서 전표번호와 계정코드 열을 선택(select)한다. (열 선택)
* 선택한 전표번호와 계정코드에서 본계정이 포함된 전표번호만 남겨서 B09 변수로 지정한다.
* 본계정과 상대계정의 전표 갯수를 세어 B09 변수로 지정한다.

Corr\_Acc = '40401'  
B09\_main <- je\_tbl |> filter(ACCTCD == Corr\_Acc) |>   
 select(JENO, ACCTCD, ACCT\_NM)  
B09\_Corr <- je\_tbl |>   
 select(JENO, ACCTCD)  
B09 <- semi\_join(B09\_Corr, B09\_main, by = 'JENO')  
B09 <- B09 |> filter(!is.na(ACCTCD)) |>   
 count(ACCTCD)

계정과목명 확인

* 계정코드와 계정과목명 열을 선택한다. 중복되는 계정과목명을 삭제하고 하나씩만 남긴다.
* 계정코드에 맞는 계정과목명을 B09 에 붙인다.

print(B09)

## # A tibble: 8 × 2  
## ACCTCD n  
## <dbl> <int>  
## 1 10800 57348  
## 2 11300 66  
## 3 25500 57216  
## 4 25900 75  
## 5 26900 2  
## 6 40100 44  
## 7 40401 57419  
## 8 54100 1

# Conclusion

전표 분석 결과 Step3에서 정의한 오류 사항이 발견되지 아니함.

| 조이회계 Joy Accounting |
| --- |
| <https://joy-accounting.netlify.com/> |

# Analysis

## 벤포드의 법칙

벤포드 분석

if(!require(benford.analysis)){install.packages("benford.analysis");library(benford.analysis)}

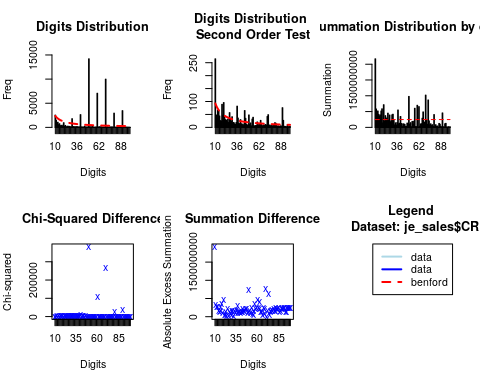
## Loading required package: benford.analysis

je\_sales <- je\_tbl |>   
 filter(ACCTCD == '40401')  
  
je\_sales$CR |> benford()

##   
## Benford object:  
##   
## Data: je\_sales$CR   
## Number of observations used = 57137   
## Number of obs. for second order = 2236   
## First digits analysed = 2  
##   
## Mantissa:   
##   
## Statistic Value  
## Mean 0.643  
## Var 0.073  
## Ex.Kurtosis 0.242  
## Skewness -1.179  
##   
##   
## The 5 largest deviations:   
##   
## digits absolute.diff  
## 1 50 13707.61  
## 2 70 9635.02  
## 3 60 6653.84  
## 4 90 3187.81  
## 5 80 2639.74  
##   
## Stats:  
##   
## Pearson's Chi-squared test  
##   
## data: je\_sales$CR  
## X-squared = 853103, df = 89, p-value < 0.00000000000000022  
##   
##   
## Mantissa Arc Test  
##   
## data: je\_sales$CR  
## L2 = 0.28011, df = 2, p-value < 0.00000000000000022  
##   
## Mean Absolute Deviation (MAD): 0.01517506  
## MAD Conformity - Nigrini (2012): Nonconformity  
## Distortion Factor: 30.68761  
##   
## Remember: Real data will never conform perfectly to Benford's Law. You should not focus on p-values!

벤포드 분석 결과 Plot 확인

bl <- je\_sales$CR |> benford()  
plot(bl)



suspectsTable

suspectsTable(bl)

## digits absolute.diff  
## 1: 50 13707.6120  
## 2: 70 9635.0184  
## 3: 60 6653.8372  
## 4: 90 3187.8062  
## 5: 80 2639.7441  
## 6: 40 2029.2705  
## 7: 16 1162.3566  
## 8: 14 1159.0087  
## 9: 19 1094.8064  
## 10: 12 1049.2025  
## 11: 17 1044.3451  
## 12: 30 985.3440  
## 13: 21 934.3609  
## 14: 23 932.0864  
## 15: 24 895.9686  
## 16: 18 891.6394  
## 17: 15 844.4772  
## 18: 26 787.4992  
## 19: 13 770.9363  
## 20: 28 762.7660  
## 21: 29 762.2427  
## 22: 25 715.2339  
## 23: 22 702.0387  
## 24: 27 688.4370  
## 25: 11 664.1250  
## 26: 33 639.7799  
## 27: 32 629.5767  
## 28: 34 624.3050  
## 29: 31 618.8212  
## 30: 37 604.7535  
## 31: 38 575.5631  
## 32: 35 569.0401  
## 33: 39 565.2433  
## 34: 42 540.8924  
## 35: 36 534.8859  
## 36: 43 523.4684  
## 37: 44 520.6478  
## 38: 41 517.9635  
## 39: 46 508.6611  
## 40: 47 476.4253  
## 41: 49 469.3157  
## 42: 53 438.8319  
## 43: 48 434.6528  
## 44: 51 424.8459  
## 45: 56 418.2023  
## 46: 57 413.5636  
## 47: 58 378.1861  
## 48: 59 368.0566  
## 49: 52 366.6673  
## 50: 66 354.1542  
## 51: 61 349.4932  
## 52: 55 341.1163  
## 53: 62 341.0370  
## 54: 64 329.7253  
## 55: 67 329.6257  
## 56: 69 324.0462  
## 57: 71 321.0586  
## 58: 68 319.2588  
## 59: 74 301.0826  
## 60: 82 294.7829  
## 61: 73 292.6145  
## 62: 76 290.3743  
## 63: 79 287.1335  
## 64: 45 286.3908  
## 65: 65 285.8514  
## 66: 85 276.2289  
## 67: 81 275.4736  
## 68: 63 271.7842  
## 69: 83 271.1806  
## 70: 87 269.5949  
## 71: 91 265.1971  
## 72: 92 257.2651  
## 73: 77 257.1887  
## 74: 88 255.3904  
## 75: 93 253.3959  
## 76: 86 251.8735  
## 77: 78 250.1098  
## 78: 20 246.6930  
## 79: 75 246.6708  
## 80: 98 245.9238  
## 81: 95 245.8378  
## 82: 84 243.6636  
## 83: 96 242.1451  
## 84: 97 235.5077  
## 85: 94 234.5875  
## 86: 99 234.3919  
## 87: 54 230.3207  
## 88: 72 223.2714  
## 89: 10 178.9461  
## 90: 89 162.2575  
## digits absolute.diff