

Example computation of 7-day low flows for climate year and summer season

Dave Lorenz

August 9, 2013

This example illustrates retrieving daily values data from NWISweb and computing the 7-day low flow values for each climate year (the year ending March 31) and summer season (beginning June and ending September). This example uses daily flow data from USGS station 05484500, Raccoon River at Van Meter, Iowa through water year 2012 (September 30, 2012).

```
> # Load the DVstats package and retrieve the data
> library(DVstats)
> # USGSwsBase is required by DVstats, so readNWIS,
> # renCol, and screenData functions are available.
> RRVM <- renCol(readNWIS("05484500", end.date="2012-09-30"))
> # The screenData function is useful to review for
> # complete record, default is by calendar year.
> with(RRVM, screenData(datetime, Flow))
```

Table of incomplete DVs:

	Month.number											
Year	1	2	3	4	5	6	7	8	9	10	11	12
1915	31	28	31	24								
1916												
1917												
1918												
1919												
1920												
1921												
1922												
1923												
1924												
1925												
1926												
1927												
1928												

1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974

```

1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012          31 30 31
Date of first value: 1915-04-25
Date of last value: 2012-09-30

> # The record is complete, beginning 1915-04-25

```

1 Compute the climate year 7-day low flows

The `dvStat` function is used to compute user-defined summary statistics for daily values. Each summary statistic requires a new call to `dvStat`. The arguments to `dvStat` include `x`, which is the daily value to be summarized; `Dates`, the date for each value in `x`; `Start` and `End`, that can be used to extract the data for a user-defined period of time; `by`, a grouping variable; `pre`, any preprocessing function; `stat`, the summary statistical function—it must return a single value and must accept the `na.rm` argument; `na.rm`, the value to use for the `na.rm` argument to `stat`; `STAID`, any name the user wants to use for the station; and any additional arguments to `pre`. For interactive use, it is easiest to use the function `with` to make specifying columns easier.

To compute climate year statistics, the `by` argument can be defined by the `climateYear` function supplying the same data as for the `Dates`.

To compute 7-day (or other) running averages, `pre` should be set to `movingAve` and its arguments `span` and `pos` should be set to 7 and "trailing" as shown in the example below.

By default, the statistic (`stat`) in `min`, which will give the low flows.

```
> # Compute the 7-day low flow for each climate year.
> RRVM7ClimY <- with(RRVM,
+   dvStat(Flow, datetime,
+     by=climateYear(datetime),
+     pre=movingAve,
+     STAID="05484500",
+     span=7, pos="trailing"))
> # print the first and last few rows of the output
> head(RRVM7ClimY)
```

	STAID	Group	Nobs	min	Date
1	05484500	1916	342	380.00000	1915-12-26
2	05484500	1917	365	53.00000	1917-01-27
3	05484500	1918	365	42.00000	1918-02-07
4	05484500	1919	365	40.85714	1918-10-09
5	05484500	1920	366	89.71429	1919-09-16
6	05484500	1921	365	289.85714	1920-10-11

```
> tail(RRVM7ClimY)
```

	STAID	Group	Nobs	min	Date
93	05484500	2008	366	414.8571	2007-08-04
94	05484500	2009	365	482.5714	2008-09-23
95	05484500	2010	365	425.0000	2009-09-23
96	05484500	2011	365	529.5714	2011-02-14
97	05484500	2012	366	176.8571	2012-01-25
98	05484500	2013	183	109.0000	2012-08-25

The data are the station id, **STaID**; the year or season, **group** as a factor; the number of observations used to compute the statistic, **Nobs**; the statistic, **min** in this case; and the date of the value, **Date**. The date of the value is the first day for the minimum average flow for the preceding 7 days,

The first and last rows are incomplete, as indicated by the number of observations that were used (**Nobs**). To use only complete climate years, the data can be subsetted by extracting only those rows where **Nobs** is greater than or equal to 365 (leap years have 366 days). Alternatively, **Start** and **End** could be set to the climate year start and end dates, but one would still need to possibly subset for incomplete years in the record.

```
> # Extract complete climate year data.
> RRVM7ClimY <- subset(RRVM7ClimY, Nobs >= 365)
> # print the first and last few rows of the output
> head(RRVM7ClimY)
```

	STaID	Group	Nobs	min	Date
2	05484500	1917	365	53.00000	1917-01-27
3	05484500	1918	365	42.00000	1918-02-07
4	05484500	1919	365	40.85714	1918-10-09
5	05484500	1920	366	89.71429	1919-09-16
6	05484500	1921	365	289.85714	1920-10-11
7	05484500	1922	365	150.00000	1922-02-07

```
> tail(RRVM7ClimY)
```

	STaID	Group	Nobs	min	Date
92	05484500	2007	365	198.2857	2006-08-09
93	05484500	2008	366	414.8571	2007-08-04
94	05484500	2009	365	482.5714	2008-09-23
95	05484500	2010	365	425.0000	2009-09-23
96	05484500	2011	365	529.5714	2011-02-14
97	05484500	2012	366	176.8571	2012-01-25

```
> # Change the name of the statistic
> names(RRVM7ClimY)[4] <- "LowQ7"
```

2 Plots and summary of the climate year data

The probability and time-series plots can be instructive. Use the `Date` column instead of `Group` because `Group` is a factor and can't be treated as a date.

```
> # setSweave is required for the vignette.  
> setSweave("SevenDay_01", 5, 5)  
> with(RRVM7ClimY, probPlot(LowQ7, yaxis.log=T, xlabels=5))  
> graphics.off()
```

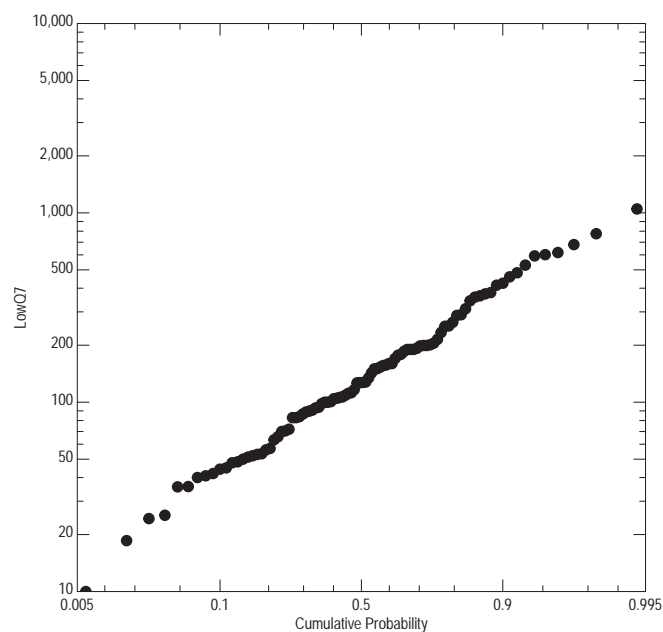


Figure 1. The probability plot.

```

> # setSweave is required for the vignette.
> setSweave("SevenDay_02", 5, 5)
> with(RRVM7ClimY, timePlot(Date, LowQ7,
+   Plot=list(what="points")))
> graphics.off()

```

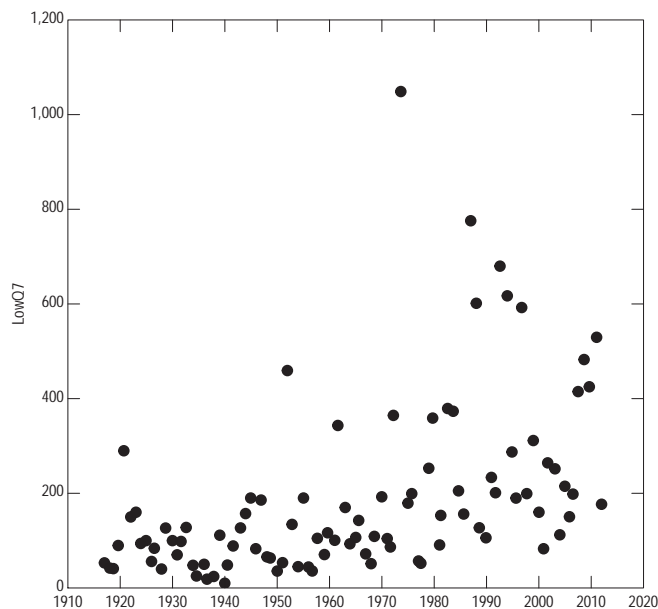


Figure 2. The time-series plot.

Figure 2 suggests a trend of increasing low-flows over time. Some applications of low-flow analysis can require a trend analysis to establish a base period. A thorough trend analysis would include an analysis of the relation of flows to precipitation and possibly land use changes and is beyond the scope of this vignette.

A useful summary statistic for annual series is a table of the number of occurrences of the lowest flows by month. For these data, most low flows occur during the winter months.

```

> with(RRVM7ClimY, table(month(Date, label=TRUE)))

```

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
25	18	1	1	1	0	2	8	16	12	0	12

3 Compute the summer season 7-day low flows

To compute seasonal statistics, the `by` argument can be defined by the `seasonYear` function supplying the same data as for the `Dates` and setting the beginning and ending months by the `start.month` and `end.month` arguments, which are "June" and "September" by default.

```
> # Compute the 7-day low flow for each climate year.
> RRVM7Summ <- with(RRVM,
+   dvStat(Flow, datetime,
+     by=seasonYear(datetime),
+     pre=movingAve,
+     STAID="05484500",
+     span=7, pos="trailing"))
> # print the first and last few rows of the output
> head(RRVM7Summ)
```

	STAID	Group	Nobs	min	Date
1	05484500	1915	122	920.71429	1915-09-08
2	05484500	1916	122	87.00000	1916-09-01
3	05484500	1917	122	188.42857	1917-09-03
4	05484500	1918	122	46.85714	1918-09-30
5	05484500	1919	122	89.71429	1919-09-16
6	05484500	1920	122	327.71429	1920-08-19

```
> tail(RRVM7Summ)
```

	STAID	Group	Nobs	min	Date
93	05484500	2007	122	414.8571	2007-08-04
94	05484500	2008	122	482.5714	2008-09-23
95	05484500	2009	122	425.0000	2009-09-23
96	05484500	2010	122	2148.5714	2010-09-18
97	05484500	2011	122	245.4286	2011-09-30
98	05484500	2012	122	109.0000	2012-08-25

The first and last rows are complete in this case because the streamflow record started in April of 1915 and continued through September of 2012. Note that there is a disconnect between the summer period year and the climate year. For example, the summer period year 2000 is actually in the 2001 climate year. This example will retain all of the data. But will rename column 4.

```
> # Change the name of the statistic
> names(RRVM7Summ)[4] <- "LowQ7"
```


4 Quick plots of the summer data

The probability and time-series plots can be instructive. Use the `Date` column instead of `Group` because `Group` is a factor and can't be treated as a date.

```
> # setSweave is required for the vignette.  
> setSweave("SevenDay_03", 5, 5)  
> with(RRVM7Summ, probPlot(LowQ7, yaxp.log=T, xlab=5))  
> graphics.off()
```

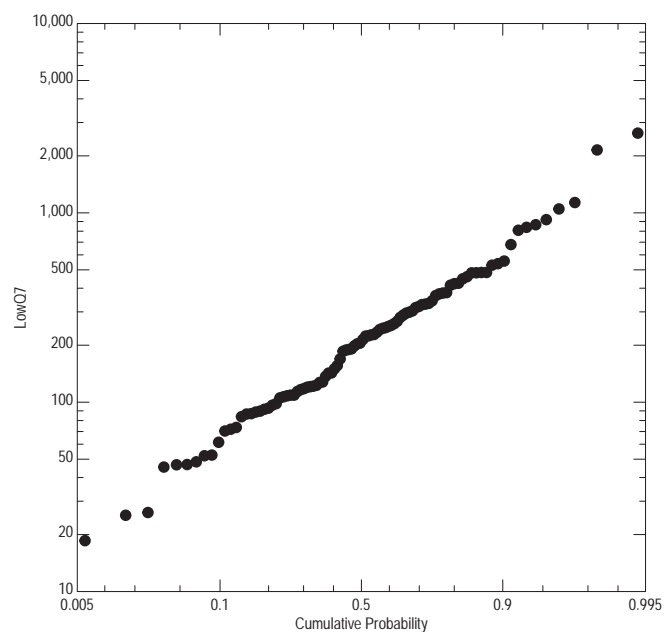


Figure 3. The probability plot of summer low flows.

```

> # setSweave is required for the vignette.
> setSweave("SevenDay_04", 5, 5)
> with(RRVM7Summ, timePlot(Date, LowQ7,
+   Plot=list(what="points")))
> graphics.off()

```

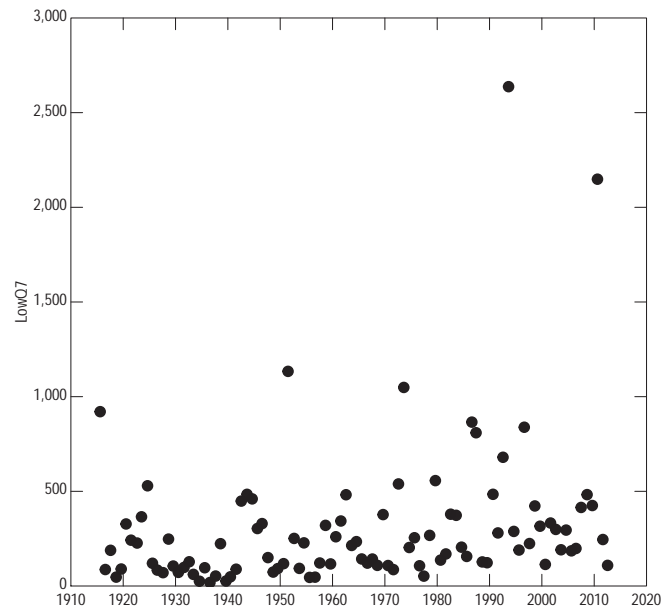


Figure 4. The time-series plot of summer low flows.

5 Compute the frequency analyses

The function `freqAnal` will perform a frequency analysis of low-flow data. It tries to fit 3 distributions—the log-Pearson type III, a 3-parameter log-normal, and the log-generalized extreme value. The log-Pearson type III distribution is fit using the method of moments—compute the mean, standard deviation, and skew of the logarithms of the data. The other distributions are fit using the method of maximum likelihood. The default distribution selection procedure preferentially selects the log-Pearson type III, then the 3-parameter log-normal, with the log-generalized extreme value as the last choice.

```
> # The Annual analysis
> RRV7ClimY.frq <- with(RRV7ClimY, freqAnal(LowQ7, id=Group,
+     desc="Annual 7-day Low Flow", STAID="05484500"))
> # The Summer analysis
> RRV7Summ.frq <- with(RRV7Summ, freqAnal(LowQ7, id=Group,
+     desc="Summer 7-day Low Flow", STAID="05484500"))
```

The respective reports are generated on the next two pages.

```
> # The Annual analysis
> print(RRVM7ClimY.frq)
```

Frequency analysis for 05484500
Statistics for Annual 7-day Low Flow

Descriptive statistics based on 96 non-zero values

1940	1937	1938	1935	1950	1957	1928	1919	1918	1956
10.00	18.57	24.29	25.29	35.71	35.86	40.00	40.86	42.00	44.29
1954	1934	1941	1936	1968	1978	1917	1951	1926	1977
45.00	47.86	48.43	50.00	51.29	52.14	53.00	53.43	56.00	57.00
1949	1948	1931	1959	1967	2001	1946	1927	1972	1942
63.29	65.71	70.00	70.57	72.00	82.86	82.86	83.86	86.57	88.71
1920	1981	1964	1924	1932	1925	1930	1961	1971	1958
89.71	90.86	93.29	94.00	98.29	100.00	100.00	100.71	104.29	104.86
1990	1965	1969	1939	2004	1960	1929	1943	1989	1933
106.00	106.71	109.00	111.43	112.29	116.57	126.57	126.86	127.14	127.86
1953	1966	1922	2006	1982	1986	1944	2000	1923	1963
134.29	142.71	150.00	150.43	153.29	156.14	157.14	160.00	160.00	170.00
2012	1975	1947	1996	1945	1955	1970	2007	1976	1998
176.86	179.14	185.71	189.86	190.00	190.00	192.57	198.29	199.29	199.29
1992	1985	2005	1991	2003	1979	2002	1995	1921	1999
201.14	205.14	214.86	233.57	251.71	252.86	264.29	287.29	289.86	311.43
1962	1980	1973	1984	1983	2008	2010	1952	2009	2011
343.29	358.86	364.57	373.29	379.14	414.86	425.00	459.29	482.57	529.57
1997	1988	1994	1993	1987	1974				
592.43	601.43	617.14	679.86	775.71	1048.71				

Sample mean: 188.9

Sample std. dev.: 181.9

Log-Pearson type III analysis.

Mean common logs = 2.114

Standard deviation common logs = 0.3831

Skewness common logs = -0.06242

PPCC:0.9969

Estimated data:

Mean: 188.1

Std. Dev.: 182.9

3-parameter Log-normal analysis.

Mean common logs = 2.123

Standard deviation common logs = 0.3733

Offset (lambda) = -1.832

PPCC:0.997

Estimated data:

Mean: 187.6

Std. Dev.: 183

Log-generalized extreme value analysis.

Location = 4.568

Scale = 0.8909

Shape = -0.3063

PPCC:0.9965

Estimated data:

Mean: 187.2

Std. Dev.: 172.5

The frequency analysis estimates will be made using the log-Pearson type III method.
No zero values in these data, no conditional adjustment needed.

```
> # The Summer analysis
> print(RRVM7Summ.frq)
```

Frequency analysis for 05484500
Statistics for Summer 7-day Low Flow

Descriptive statistics based on 98 non-zero values

1936	1934	1939	1955	1956	1918	1940	1977	1937	1933
18.57	25.29	26.14	45.43	46.71	46.86	48.43	52.14	52.57	61.43
1927	1930	1948	1926	1971	1916	1941	1919	1949	1953
70.43	72.00	73.57	83.86	86.57	87.00	88.71	89.71	91.71	93.00
1935	1931	1929	1976	1970	2012	1968	2000	1959	1950
96.43	98.29	105.43	106.86	108.14	109.00	109.00	114.00	116.57	118.00
1925	1966	1957	1989	1988	1932	1980	1967	1965	1947
120.00	120.86	121.57	122.86	127.14	127.86	136.86	142.43	142.71	150.00
1985	1981	2005	1917	1995	2003	2006	1974	1984	1963
156.14	169.29	185.71	188.43	189.86	191.29	198.29	203.29	205.14	214.14
1938	1997	1922	1954	1964	1921	2011	1928	1952	1975
223.43	224.14	226.86	227.86	234.29	242.00	245.43	247.71	251.29	254.86
1960	1978	1991	1994	2004	2002	1945	1999	1958	1920
260.00	267.43	280.29	288.29	295.29	298.86	303.57	315.86	320.29	327.71
1946	2001	1961	1923	1983	1969	1982	2007	1998	2009
329.57	332.57	343.29	365.43	373.29	377.14	379.14	414.86	422.71	425.00
1942	1944	1962	2008	1943	1990	1924	1972	1979	1992
448.43	460.14	482.00	482.57	484.14	484.29	529.29	539.00	556.57	679.86
1987	1996	1986	1915	1973	1951	2010	1993		
809.43	838.57	865.57	920.71	1048.71	1133.57	2148.57	2637.14		

Sample mean: 308.2

Sample std. dev.: 378.7

Log-Pearson type III analysis.

Mean common logs = 2.299

Standard deviation common logs = 0.4004

Skewness common logs = 0.08179

PPCC:0.996

Estimated data:

Mean: 302.4

Std. Dev.: 330.4

3-parameter Log-normal analysis.

Mean common logs = 2.293

Standard deviation common logs = 0.4044

Offset (lambda) = 1.968

PPCC:0.9958

Estimated data:

Mean: 299.7

Std. Dev.: 317.8

Log-generalized extreme value analysis.

Location = 4.951

Scale = 0.9087

Shape = -0.2379

PPCC:0.9946

Estimated data:

Mean: 308

Std. Dev.: 336.4

The frequency analysis estimates will be made using the log-Pearson type III method. No zero values in these data, no conditional adjustment needed.

In both cases, the log-Pearson type III distribution was selected. The probability plot correlation coefficient is very close to 1, suggesting a very good fit. The `plot` function can be used to assess the selected fit.

```

> # The Annual analysis
> setSweave("SevenDay_05", 5, 5)
> plot(RRVM7ClimY.frq, which="default", set.up=FALSE)
> graphics.off()

```

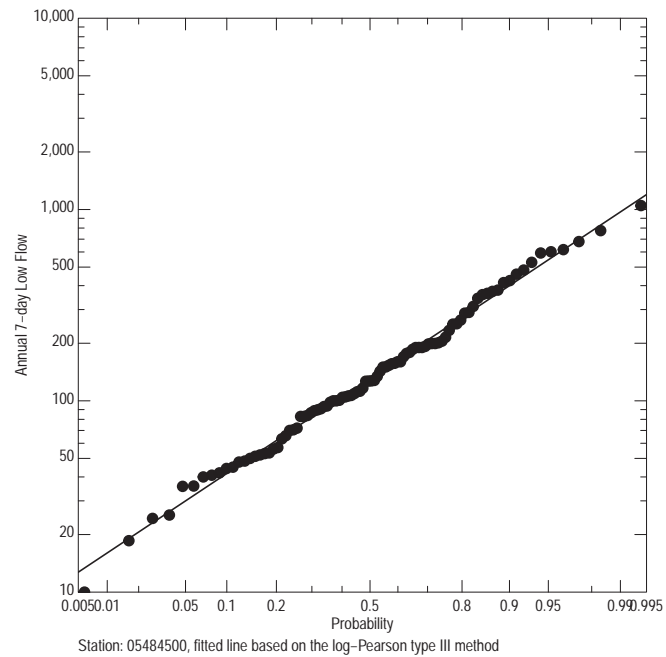


Figure 5. The log-Pearson type III fit to the annual data.


```

> # The Annual analysis
> setSweave("SevenDay_06", 5, 5)
> plot(RRVM7Summ.frq, which="default", set.up=FALSE)
> graphics.off()

```

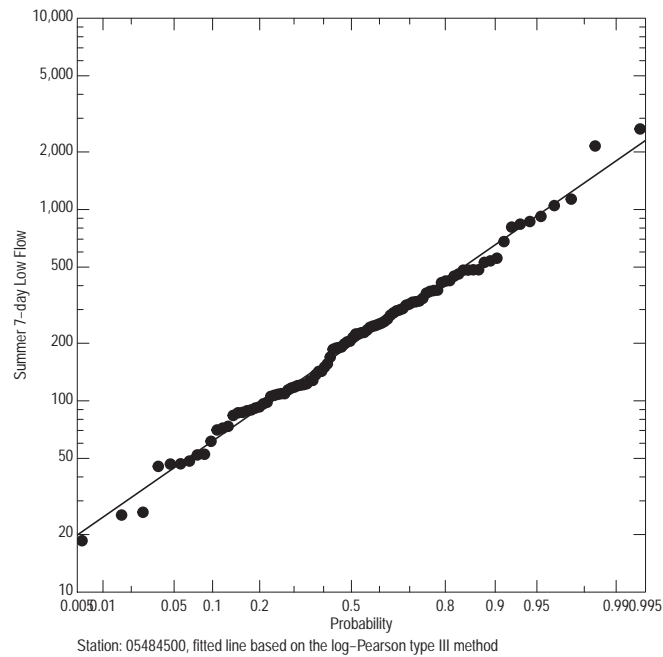


Figure 6. The log-Pearson type III fit to the summer data.

The fit is very good in both cases. The more traditional low-flow graphs (on a normal probability scale) and estimated values for selected annual or seasonal probabilities are shown on the final two pages.

```
> # The Annual analysis
> predict(RRVM7ClimY.frq)
```

	Probs	Est
1	0.01	16.02797
2	0.02	20.61116
3	0.05	29.97845
4	0.10	41.71010
5	0.20	62.01605
6	0.50	131.13956
7	0.80	273.72654
8	0.90	400.06818
9	0.96	597.32369
10	0.98	772.22369
11	0.99	971.51574

```
> setSweave("SevenDay_07", 5, 5)
> plot(RRVM7ClimY.frq, which="default", set.up=FALSE)
> graphics.off()
```

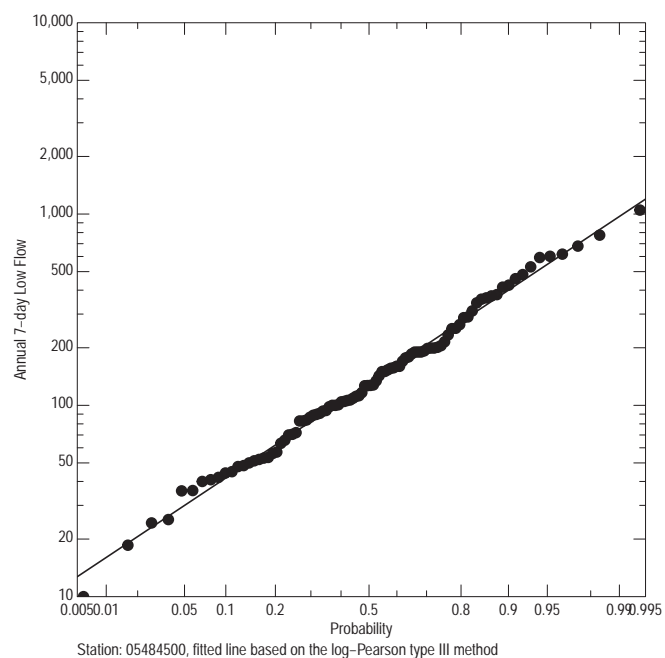


Figure 7. The log-Pearson type III estimates for the annual data.

```
> # The Summer analysis
> predict(RRVM7Summ.frq)
```

	Probs	Est
1	0.01	24.65300
2	0.02	31.22726
3	0.05	44.67016
4	0.10	61.61250
5	0.20	91.34380
6	0.50	196.64310
7	0.80	430.93313
8	0.90	654.05639
9	0.96	1026.14193
10	0.98	1376.76497
11	0.99	1797.02664

```
> setSweave("SevenDay_08", 5, 5)
> plot(RRVM7Summ.frq, set.up=FALSE)
> graphics.off()
```

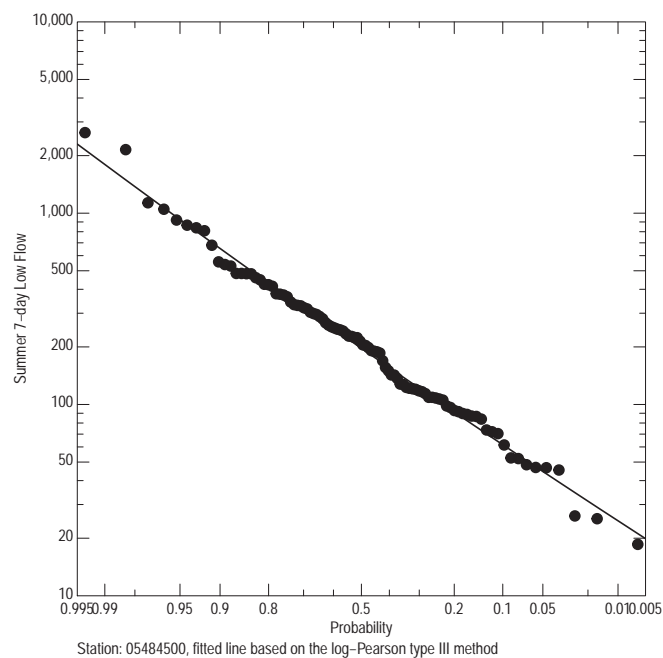


Figure 8. The log-Pearson type III estimates for the summer data.