Thomas Pattara
SAS homework 4

1. In studying the distribution of primes, a variable of interest is the so-called prime gap. It is the difference between two consecutive prime numbers. This distribution has been studied extensively in number theory. For example, the first six prime numbers are 2, 3, 5, 7, 11, and 13. And the associated prime gaps are 1, 2, 2, 4, 2. You can generate this variable by first creating a variable containing the previous prime numbers using the LAG function (see Section 11.11 of the textbook).
1a) Using SAS, obtain the prime gap distribution for prime numbers less than or equal to 500, 000. Hint: Use PROC FREQ to get the frequency distribution of the gaps.
1b) Use PROC GCHART to obtain the bar-chart associated with the frequency table in Problem 1a. You may modify the following code:

```
proc gchart data = XXXX;
vbar ZZZZ / discrete type=percent;
run;
```

1c) Find the first twenty consecutive pairs of prime numbers with a prime gap of 2. These are also called twin primes.

```
data primes;
length status $12.;

do i=2 to 500000;
   status='Prime';
   do j=2 to i-1;
      if mod(i, j) = 0 then do;
         status='Composite';
         leave; *exit loop;
      end;
   end;
output;
end;
run;

data diff;
set primes;
if status = 'Prime';
diff = i-lag(i);

if diff = 2 then twin = i ;

proc print data= diff;
  var status i diff;
```

```
  *where status eq 'Prime';
  run;

title "Gap Frequencies";
proc freq data=diff;
tables diff;
run;

proc gchart data = diff;
vbar diff / discrete type=percent;
run;


proc print data = diff(obs=65);
var i twin;

run;
```

RESULTS are in the pdf file uploaded with this document.

2. Using appropriate DO loop, find the smallest pairs of primes with a prime gap of 100. Note: Your search may go beyond 500,000.

```
data primes;
length status $12.;

do i=2 to 500000;
   status='Prime';
   do j=2 to i-1;
      if mod(i, j) = 0 then do;
         status='Composite';
         leave; *exit loop;
      end;
   end;
output;
end;
run;

data diff;
set primes;
if status = 'Prime';
```

```
diff = i-lag(i);

if diff = 100 then twin = i ;

proc print data= diff noobs;
  var status i diff;
  *where status eq 'Prime';
  run;

proc print data = diff noobs;;
var i twin;

run;
```

status	i	diff
Prime	396703	.
Prime	396709	6
Prime	396713	4
Prime	396719	6
Prime	396733	14
Prime	396833	100
Prime	396871	38
Prime	396881	10
Prime	396883	2
Prime	396887	4

i	twin
396703	.
396709	.

i	twin
396713	.
396719	.
396733	.
396833	396833
396871	.
396881	.
396883	.
396887	.

3. Create another variable, CuPRIME, that contains the number of primes less than or equal to a given number n. For n in steps of 1000, i.e., 1000, 2000, 3000, ..., 1,000,000 find CuPRIME. In number theory, this number is usually approximated by n/(log n). Calculate the true and approximate values using SAS loop and compare them. Also obtain the plot of one versus the other (put CuPRIME on the horizontal axis). You may use PROC GPLOT procedure.

```
data primes;
length status $12.;

do i=2 to 1000;
   status='Prime';


   do j=2 to i-1;
      if mod(i, j) = 0 then do;
         status='Composite';
         leave; *exit loop;

      end;
   end;
output;
end;
run;
```

```
data diff;
set primes;
if status = 'Prime';
diff = i-lag(i);

if diff = 2 then twin = i ;
n=1000/log(1000);
run;

data new;
  set diff (drop= j);
  by status;
  if first.status then count=0;
  count+1;
  if last.status then output;
run;

proc print;
run;
```

Obs	status	i	diff	twin	n	count
1	Prime	997	6	.	144.765	168