

HW4_SarahWidener

Widener

9/20/2020

R Markdown

```
library("tidyverse")
```

```
## -- Attaching packages ----- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2    v purrr  0.3.4
## v tibble  3.0.3    v dplyr  1.0.2
## v tidyr   1.1.2    v stringr 1.4.0
## v readr   1.3.1    v forcats 0.5.0
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
callfile <- read.delim('~/.HW4/HW4/pheno.txt')
#str(callfile)
```

- 1) How many unique observations are there in the “GENO” variable? (You haven’t used this function in class yet. Try to look for functions.)

```
length(unique(callfile$GENO))
```

```
## [1] 485
```

- 2) Save as a text file a subset of the pheno data set containing only range 1.

```
subset(callfile, callfile$range == 1)
```

##	LOC	GENO	TAXA	range	row	BLOCK	Y	M	h1
## 1	12EF	Mila	PI506058	1	1	1	13.245637	0.6210611	0.210
## 2	12EF	PRE0465	PI155138	1	2	1	12.253702	0.6716969	0.264
## 3	12EF	PRE1022	PI329902	1	3	1	4.241176	0.7613764	0.265
## 4	12EF	PRE0139	PI276790	1	4	1	7.478768	0.7423784	0.247
## 5	12EF	PRE0237	PI570719	1	5	1	4.498335	0.7550720	0.321
## 6	12EF	PRE0028	PI148089	1	6	1	9.348920	0.7644761	0.298
## 7	12EF	PRE0049	PI152595	1	7	1	NA	NA	0.237

## 8	12EF	E105	PI297171	1	8	1	9.008476	0.6896845	0.236
## 9	12EF	PRE0500	NSL55745	1	9	1	7.044026	0.7950060	0.223
## 10	12EF	PRE0006	PI147837	1	10	1	NA	NA	0.228
## 11	12EF	PRE0016	PI147933	1	11	1	3.927760	0.7713900	0.230
## 12	12EF	ZMA_5356	PI505737	1	12	1	5.213096	0.6480313	0.273
## 13	12EF	PRE0240	NSL55665	1	13	2	11.493525	0.6040818	0.249
## 14	12EF	PRE0212	PI291237	1	14	2	13.976343	0.6068208	0.291
## 15	12EF	PRE0069	PI153824	1	15	2	2.354613	0.7908228	0.205
## 16	12EF	PRE0196	PI248313	1	16	2	8.722704	0.5793399	0.249
## 17	12EF	PRE1211	NSL102183	1	17	2	3.954825	0.7329851	0.216
## 18	12EF	PRE0671	NSL51603	1	18	2	13.196102	0.5632592	0.270
## 19	12EF	PRE0042	NSL52312	1	19	2	6.964302	0.7061228	0.193
## 20	12EF	PRE0704	NSL51876	1	20	2	7.570572	0.6349024	0.250
## 21	12EF	ZMA_5376	PI505740	1	21	2	10.126498	0.6712966	0.267
## 22	12EF	PRE0068	PI153800	1	22	2	3.701685	0.7500761	0.245
## 23	12EF	PRE0041	PI148114	1	23	2	11.083699	0.7121809	0.270
## 24	12EF	PRE1022	PI329902	1	24	2	3.375241	0.7626212	0.241
## 385	13EF	PRE0356	NSL51981	1	1	1	8.997881	0.7591522	0.200
## 386	13EF	PRE0146	PI276801	1	4	1	18.894927	0.6455331	0.300
## 387	13EF	PRE0140	PI276791	1	5	1	8.524516	0.7401316	0.200
## 388	13EF	PRE0244	NSL50401	1	6	1	9.036460	0.7746114	0.175
## 389	13EF	PRE0277	PI562969	1	7	2	17.047332	0.5748032	0.250
## 390	13EF	PRE0212	PI291237	1	8	2	10.661660	0.7074830	0.300
## 391	13EF	PRE1446	PI505705	1	9	2	11.034716	0.7247706	0.225
## 392	13EF	PRE0025	PI148086	1	10	2	15.347401	0.7627737	0.225
## 393	13EF	PRE0382	PI660597	1	11	2	7.750967	0.7974684	0.200
## 394	13EF	PRE0373	PI660602	1	12	2	11.372848	0.7576471	0.250
## 395	13EF	Pacesetter commercial_hybrid		1	2	1	NA	0.6977401	0.300
## 396	13EF	PRE1321	NSL365751	1	3	1	8.363899	0.7485207	0.300
## 865	14EF	PRE1116	NSL51949	1	1	1	4.747282	0.8346056	0.400
## 866	14EF	PRE0590	NSL55749	1	4	1	7.279855	0.8074866	0.390
## 867	14EF	PRE0337	PI562944	1	5	1	6.474841	0.7846715	0.325
## 868	14EF	PRE0725	NSL51906	1	6	1	5.795618	0.6466431	0.385
## 869	14EF	ZMA_5293	PI505720	1	7	2	10.048599	0.7909408	0.525
## 870	14EF	PRE1125	NSL50971	1	8	2	6.770767	0.7935943	0.350
## 871	14EF	PRE0600	PI152862	1	9	2	9.081728	0.6810631	0.475
## 872	14EF	PRE0784	NSL54496	1	10	2	3.573682	0.8365854	0.350
## 873	14EF	PRE0559	NSL51693	1	11	2	9.276392	0.7422680	0.425
## 874	14EF	PRE1156	PI330098	1	12	2	8.243323	0.7377778	0.325
## 875	14EF	Pacesetter commercial_hybrid		1	2	1	13.292800	0.7647059	0.595
## 876	14EF	PRE0156	PI276817	1	3	1	4.667122	0.8048780	0.425
##	h2	h3	h4						
## 1	1.250	2.300	3.300						
## 2	1.250	2.200	3.200						
## 3	1.150	2.050	2.400						
## 4	1.400	2.450	3.700						
## 5	1.200	1.950	2.650						
## 6	1.300	2.550	3.750						
## 7	0.800	1.950	2.150						
## 8	0.950	2.150	3.350						
## 9	0.950	1.850	2.600						
## 10	0.750	1.850	2.850						
## 11	0.950	2.300	3.100						
## 12	1.250	2.400	3.550						

```
## 13  1.400 2.300 3.400
## 14  1.400 2.350 3.550
## 15  1.000 1.750 2.250
## 16  1.500 2.550 3.750
## 17  0.950 1.850 2.550
## 18  1.500 2.500 3.400
## 19  0.900 1.750 2.150
## 20  1.500 2.350 3.450
## 21  1.350 2.050 3.450
## 22  1.250 2.150 3.100
## 23  1.350 2.350 3.350
## 24  1.200 1.850 2.900
## 385 1.500 2.600 3.000
## 386 1.850 3.400 3.900
## 387 1.500 3.900 5.000
## 388 1.300 2.600 3.300
## 389 1.700 3.200 3.800
## 390 1.550 3.300 3.900
## 391 1.200 3.000 3.800
## 392 1.750 3.700 4.100
## 393 1.150 2.700 3.300
## 394 1.800 3.600 4.200
## 395 1.900 3.300 3.500
## 396 1.250 2.250 3.100
## 865 1.035 3.086 3.297
## 866 1.100 2.646 3.493
## 867 0.750 2.262 2.898
## 868 1.200 3.092 3.073
## 869 1.270 3.008 3.667
## 870 0.860 2.577 3.110
## 871 1.735 4.428 4.010
## 872 1.205 3.155 3.092
## 873 1.350 3.107 3.704
## 874 1.390 3.784 4.250
## 875 1.440 3.421 3.601
## 876 1.070 1.770 2.281
```

```
write.table(subset(callfile, callfile$range == 1), "Sarah")
```

3) The range (minimum and maximum value) of the “Y”, “M”, and “h4” variables.

```
(range_Y <- range(callfile$Y, na.rm = TRUE))
```

```
## [1]  0.5250602 21.6188835
```

```
(range_M <- range(callfile$M, na.rm = TRUE))
```

```
## [1] 0.4502466 0.8779443
```

```
(range_h4 <- range(callfile$h4, na.rm = TRUE))
```

```
## [1] 1.204 5.100
```

4) What row contains the maximum value of “h4”?

```
maxvalueh4 <-which(callfile$h4 == max(callfile$h4, na.rm = TRUE))
maxvalueh4
```

```
## [1] 427 444 576
```

```
maxvaluerow <- which(callfile$h4 == maxvalueh4)
maxvaluerow
```

```
## integer(0)
```

```
exactrow <- callfile[maxvaluerow,5]
exactrow
```

```
## integer(0)
```

5) Create a loop that prints the mean of the variables “Y”, “M”, “h1”, “h2”, “h3”, and “h4” (it must be done using a loop, e.g., for, while, or repeat).

```
abc <- callfile[,c("Y", "M", "h1", "h2","h3","h4")]
names<- variable.names(abc)
for (i in names) {
  x<- mean(abc[,i],na.rm = TRUE)
  print(x)
}
```

```
## [1] 7.99446
## [1] 0.7431033
## [1] 0.2766773
## [1] 1.248048
## [1] 2.621128
## [1] 3.165503
```

6) Create a dataset called pheno2 and replace the values of “Y” that are smaller than 2 by 0.

```
pheno2<-callfile
pheno2$Y[pheno2$Y<2]<-0
head(pheno2)
```

```
##   LOC   GENO   TAXA range row BLOCK      Y      M   h1   h2   h3
## 1 12EF   Mila PI506058    1   1     1 13.245637 0.6210611 0.210 1.25 2.30
## 2 12EF PRE0465 PI155138    1   2     1 12.253702 0.6716969 0.264 1.25 2.20
## 3 12EF PRE1022 PI329902    1   3     1  4.241176 0.7613764 0.265 1.15 2.05
## 4 12EF PRE0139 PI276790    1   4     1  7.478768 0.7423784 0.247 1.40 2.45
## 5 12EF PRE0237 PI570719    1   5     1  4.498335 0.7550720 0.321 1.20 1.95
## 6 12EF PRE0028 PI148089    1   6     1  9.348920 0.7644761 0.298 1.30 2.55
##      h4
## 1 3.30
## 2 3.20
## 3 2.40
## 4 3.70
## 5 2.65
## 6 3.75
```