CISC HW6

Question1

> Download the baby name data for 2016 (https://storm.cis.fordham.edu/ji/cisc5500/yob2016.txt) . Write a R script that reads this file and plots a bar chart of the 10 most popular names regardless of gender, namely the x-axis shows the 10 names and the y-axis shows the number of babies who have a given name.

Preprocessing

In the first part, the script read the text file as the input with the edited column names. After the re-order with both two genders, it select the top 10 rows as the output.

```
df <- read.csv('/Users/lordxuzhiyu/Desktop/yob2016.txt', col.names = c('Name', 'Sex', 'Count'))
res <- df[with(df, order(-Count)),]
result <- head(res, 10)
result
```

	Name <fctr>	**Sex** <fctr>	**Count** <int>
1	Olivia	F	19246
18757	Noah	M	19015
18758	Liam	M	18138
2	Ava	F	16237
3	Sophia	F	16070
18759	William	M	15668
18760	Mason	M	15192
18761	James	M	14776
4	Isabella	F	14722
18762	Benjamin	M	14569

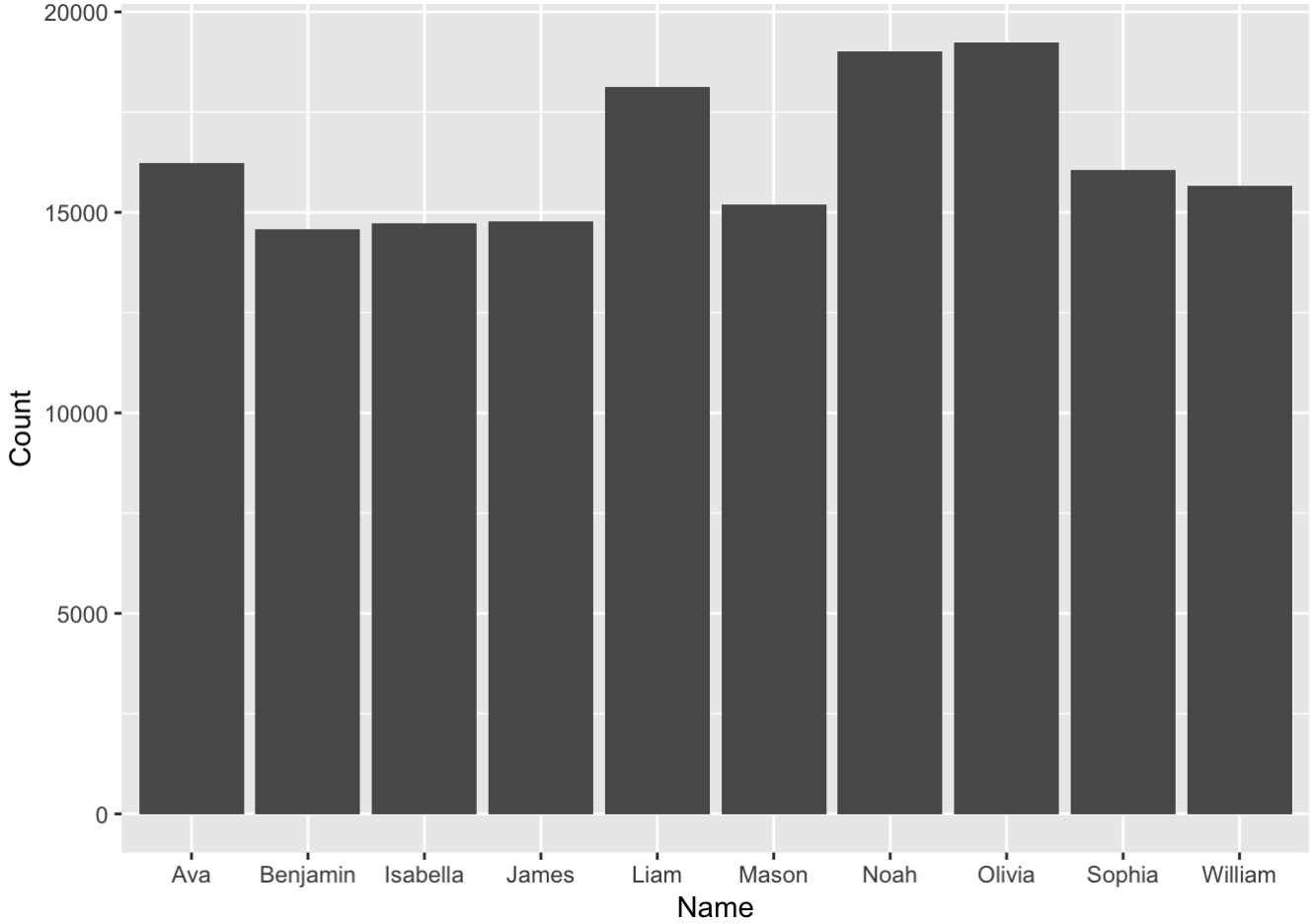
1-10 of 10 rows

Plot with ggplot2

After that, we use the ggplot2 package to draw the bar chart. The result is below:

```
library(ggplot2)

ggplot(result, aes(x = Name, y = Count)) + geom_bar(stat = "identity")
```



Question 2

> Write a R function that takes a vector (assuming the elements are integers) and returns a vector of only the elements that are multiple of 4 but not multiple of 100.

Design

We initalize a vector with a range of integers. Then we just output the result with the condition we want.

```
input <- 1:1000
input[((input %% 4) == 0)&((input %% 100) != 0)]
```

```
##   [1]   4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68
##  [18]  72  76  80  84  88  92  96 104 108 112 116 120 124 128 132 136 140
##  [35] 144 148 152 156 160 164 168 172 176 180 184 188 192 196 204 208 212
##  [52] 216 220 224 228 232 236 240 244 248 252 256 260 264 268 272 276 280
##  [69] 284 288 292 296 304 308 312 316 320 324 328 332 336 340 344 348 352
##  [86] 356 360 364 368 372 376 380 384 388 392 396 404 408 412 416 420 424
## [103] 428 432 436 440 444 448 452 456 460 464 468 472 476 480 484 488 492
## [120] 496 504 508 512 516 520 524 528 532 536 540 544 548 552 556 560 564
## [137] 568 572 576 580 584 588 592 596 604 608 612 616 620 624 628 632 636
## [154] 640 644 648 652 656 660 664 668 672 676 680 684 688 692 696 704 708
## [171] 712 716 720 724 728 732 736 740 744 748 752 756 760 764 768 772 776
## [188] 780 784 788 792 796 804 808 812 816 820 824 828 832 836 840 844 848
## [205] 852 856 860 864 868 872 876 880 884 888 892 896 904 908 912 916 920
## [222] 924 928 932 936 940 944 948 952 956 960 964 968 972 976 980 984 988
## [239] 992 996
```