the report for store in CA

Firstly, let's read the data from computer.

```
setwd("~/Documents/")

datadata=read.csv('srp1001.csv')
library(ggplot2)
```

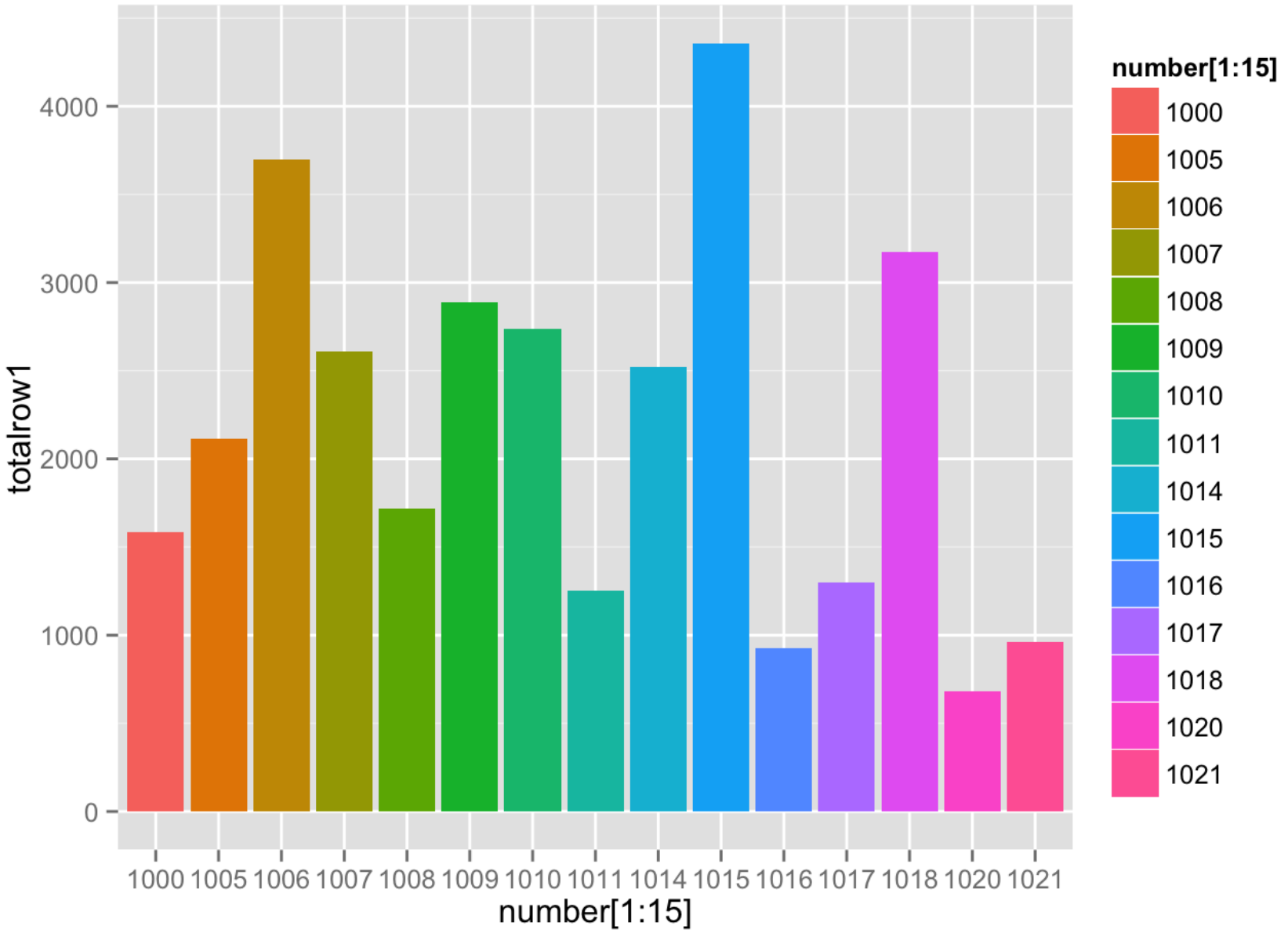
```
## Warning: package 'ggplot2' was built under R version 3.1.3
```

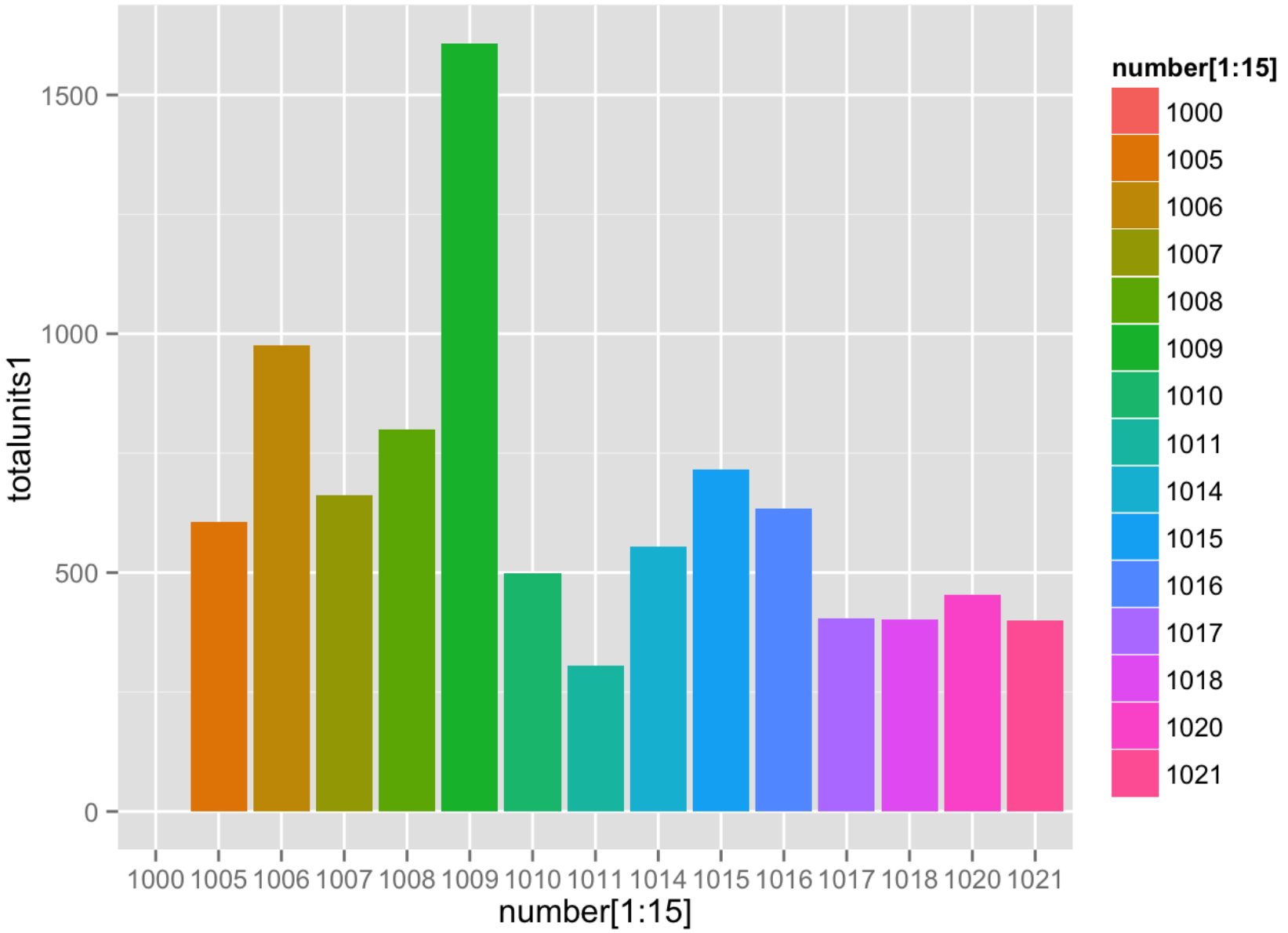
```
data8=data.frame(datadata$SITE.NUMBER,datadata$DIM,datadata$ATTRIBUTE,datadata$TOTAL.UNITS,datadata$BEG.OCC.U
NITS,datadata$VACANT.UNITS,datadata$NEW.RENTALS,datadata$STREET.RATE,datadata$Year,datadata$Month)
names(data8)=c('SITE.NUMBER','DIM','ATTRIBUTE','TOTAL.UNITS','BEG','VACANT.UNITS','NEW.RENTALS','STREET.RATE'
,'Year','Month')
```

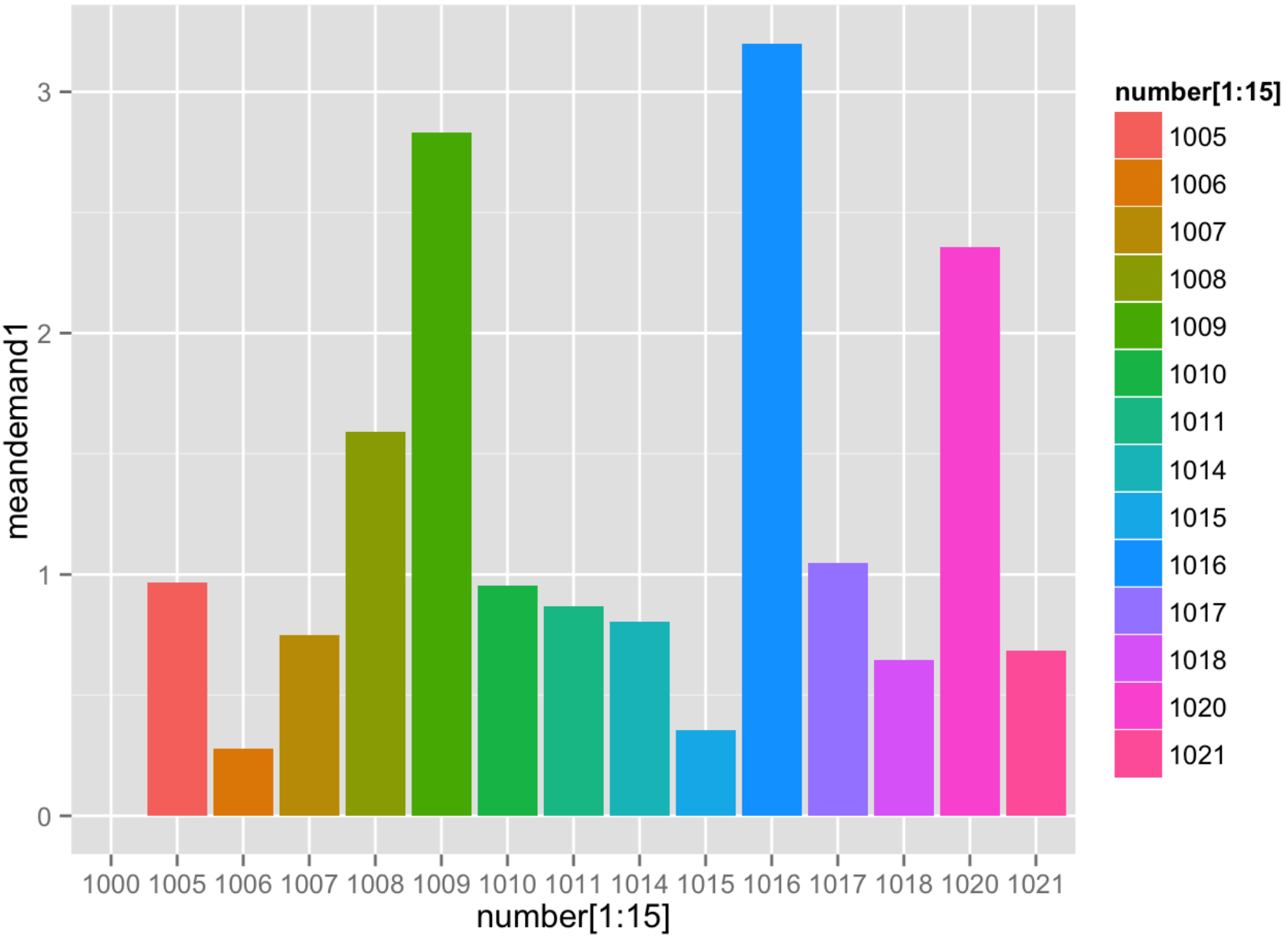
```
newdata2=read.csv('srp666.csv')
newdata2clean=newdata2[-which(newdata2$NEW.RENTALS==0&newdata2$TOTAL.UNIT==newdata2$BEG),]

write.csv(newdata2clean,'srp777.csv')
n201302=newdata2[which(newdata2$Month=='2'&newdata2$Year==2013),]

dataclean1=newdata2clean[-c(as.numeric(row.names(n201302[0:3280,])),as.numeric(row.names(n201303[0:3284,]))),
]
```

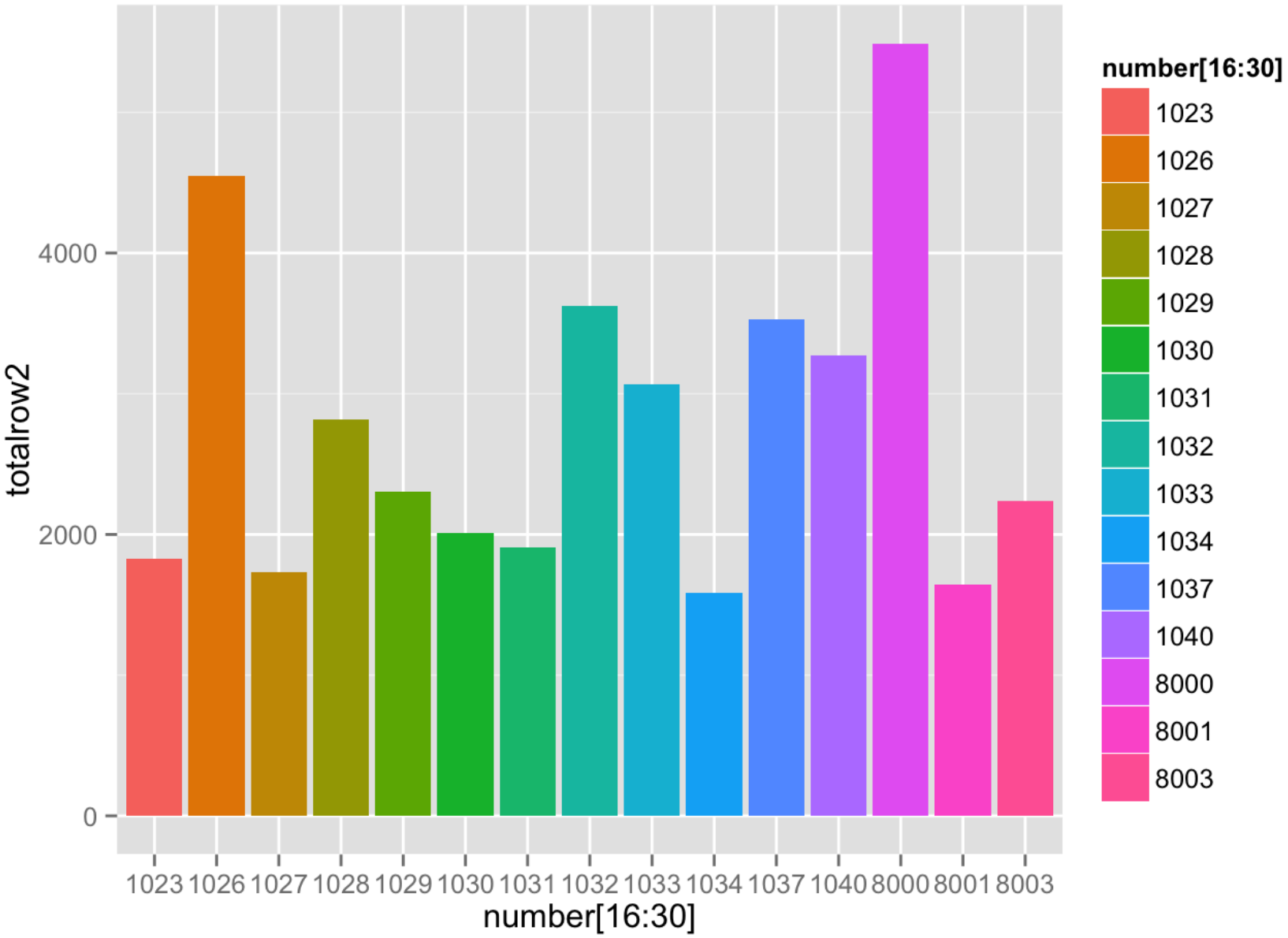
```
write.csv(dataclean1,'srp888.csv')

n201303=newdata2[which(newdata2$Month==3&newdata2$Year==2013),]

d1001=read.csv('srp888.csv')
d1001=d1001[which(d1001$SITE.NUMBER=='1001'),]



number=unique(newdata2$SITE.NUMBER[which(newdata2$state=='CA')])
number[51]=9075
number[57]=1011
number[58]=9076
```

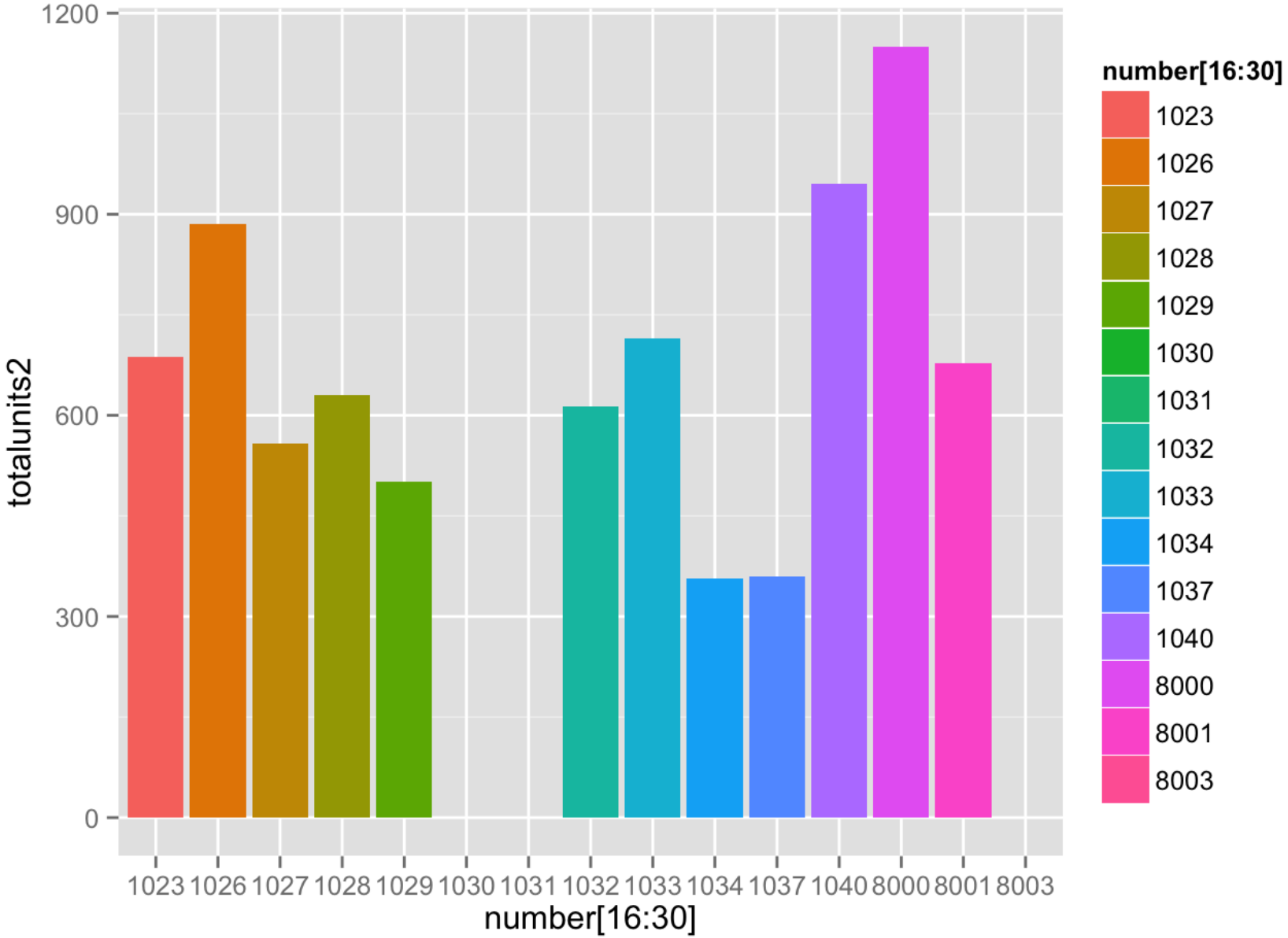
we would like to see the total rows of data for each store.

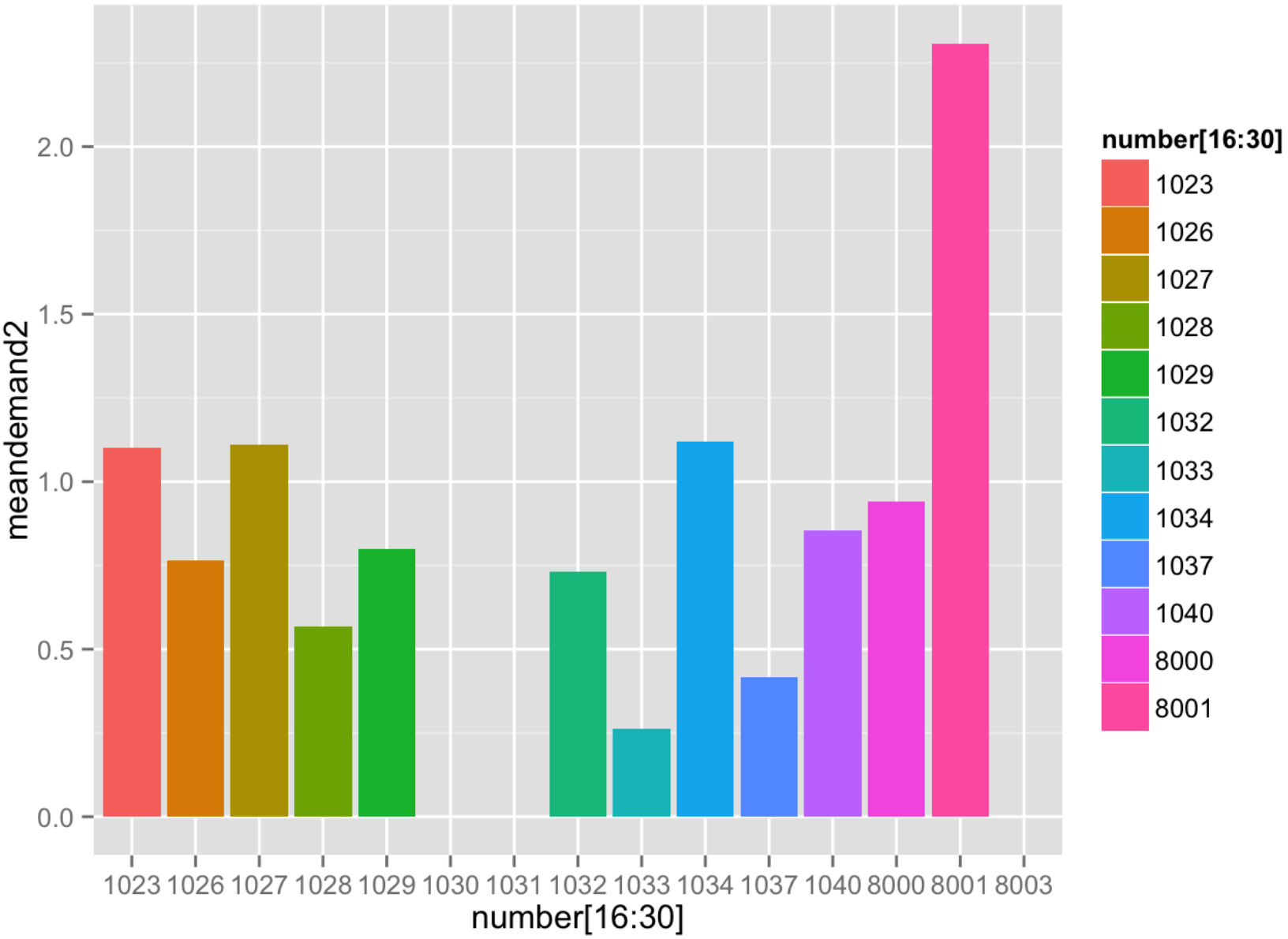


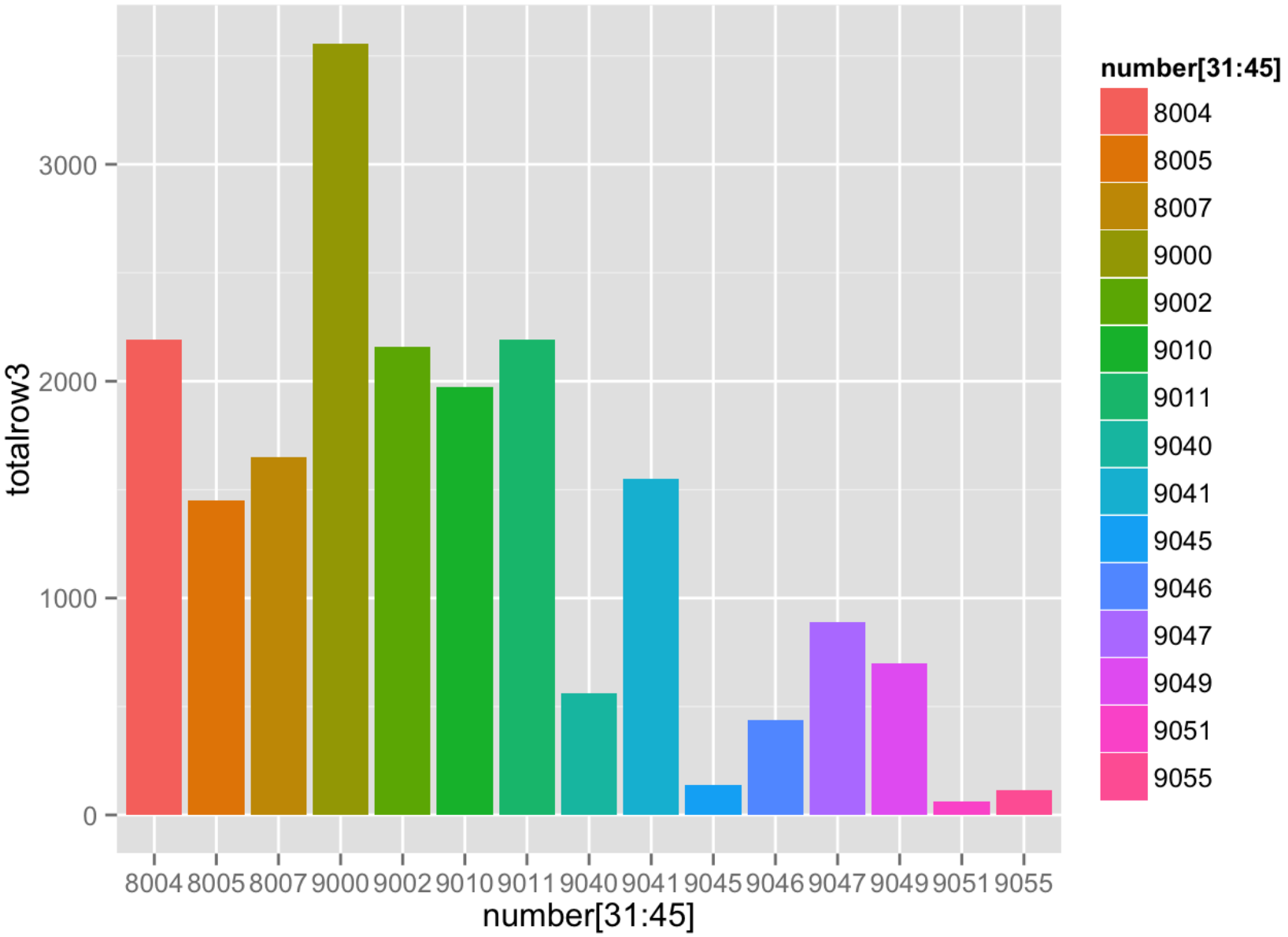


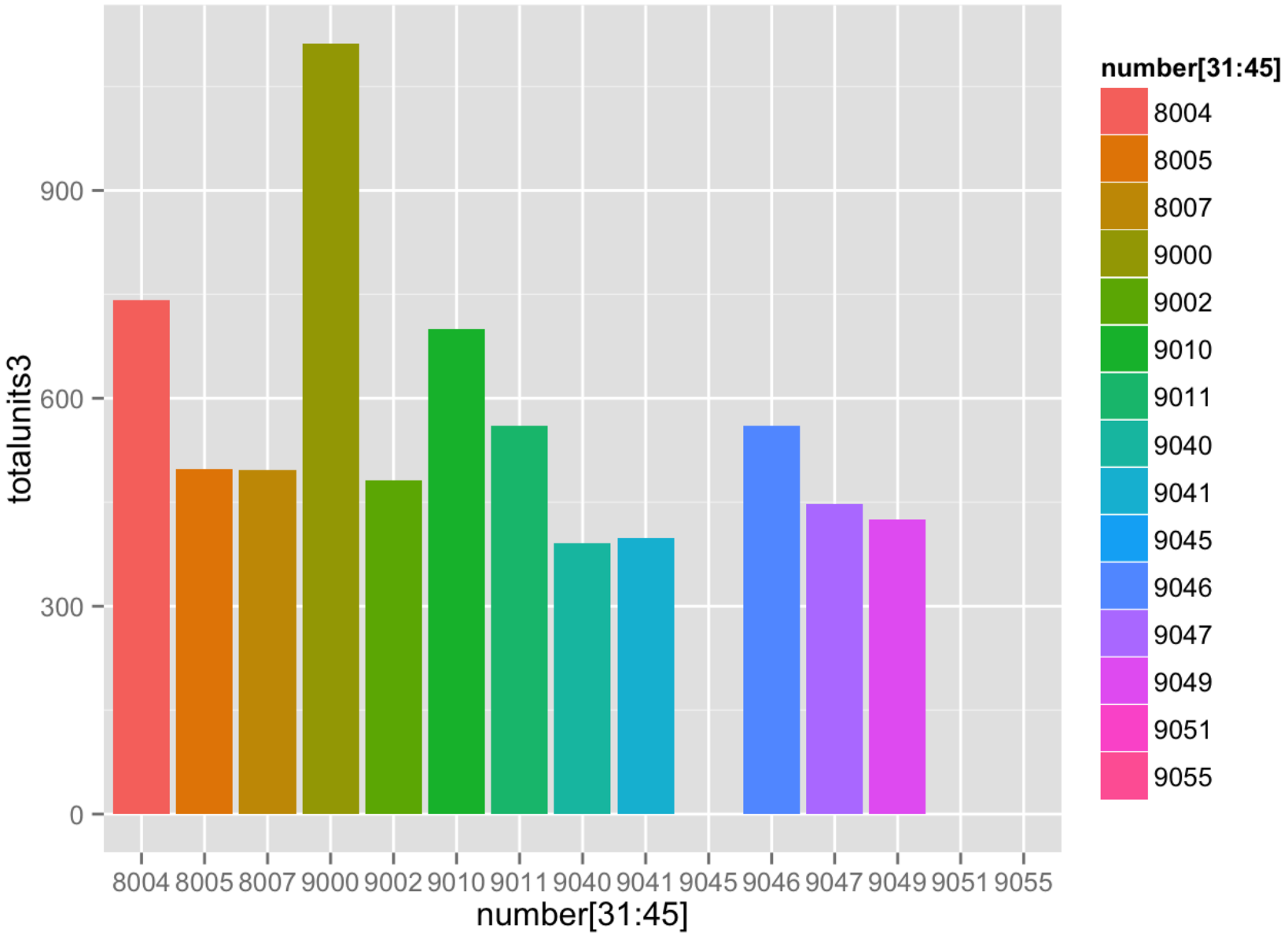


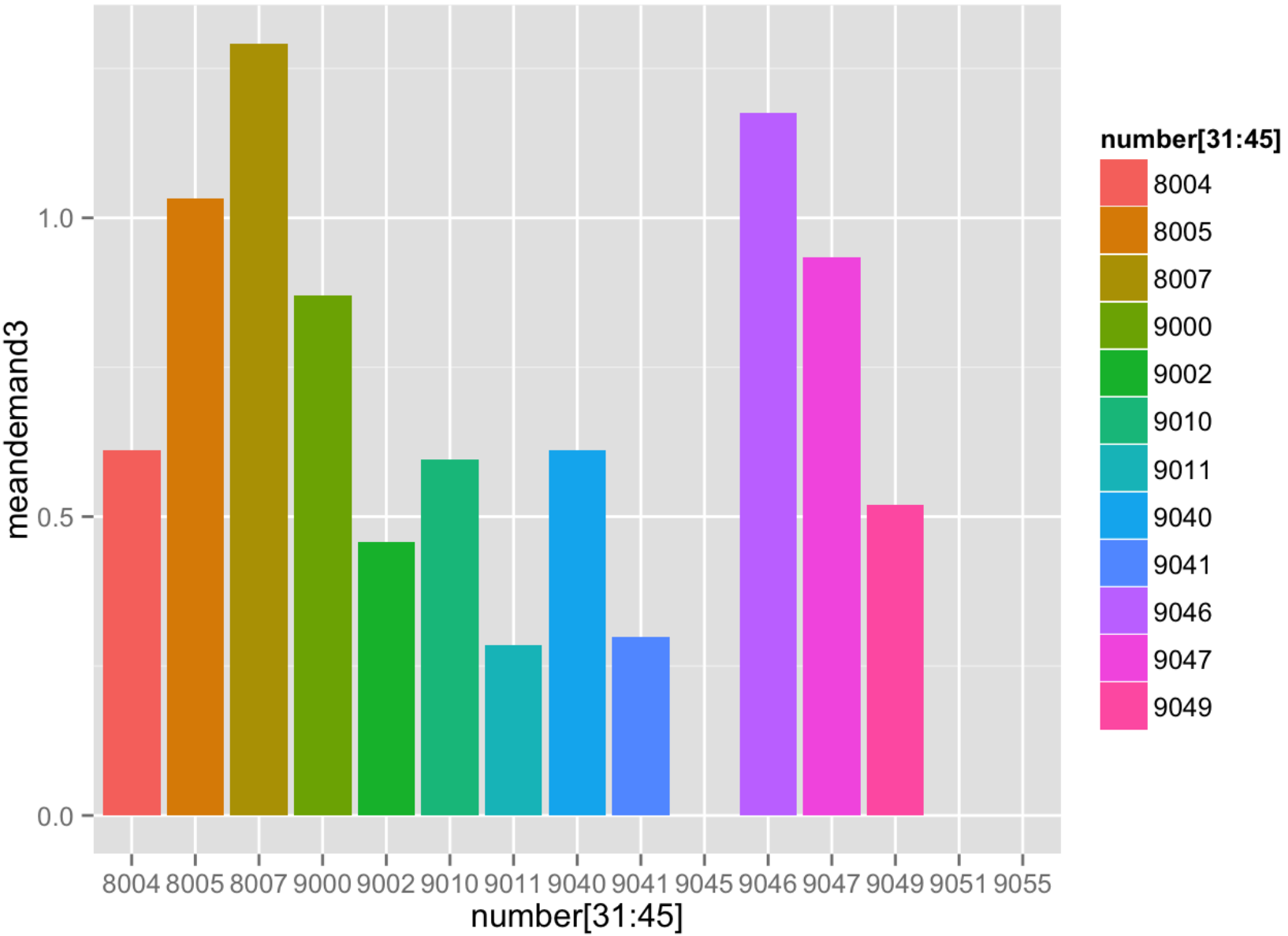


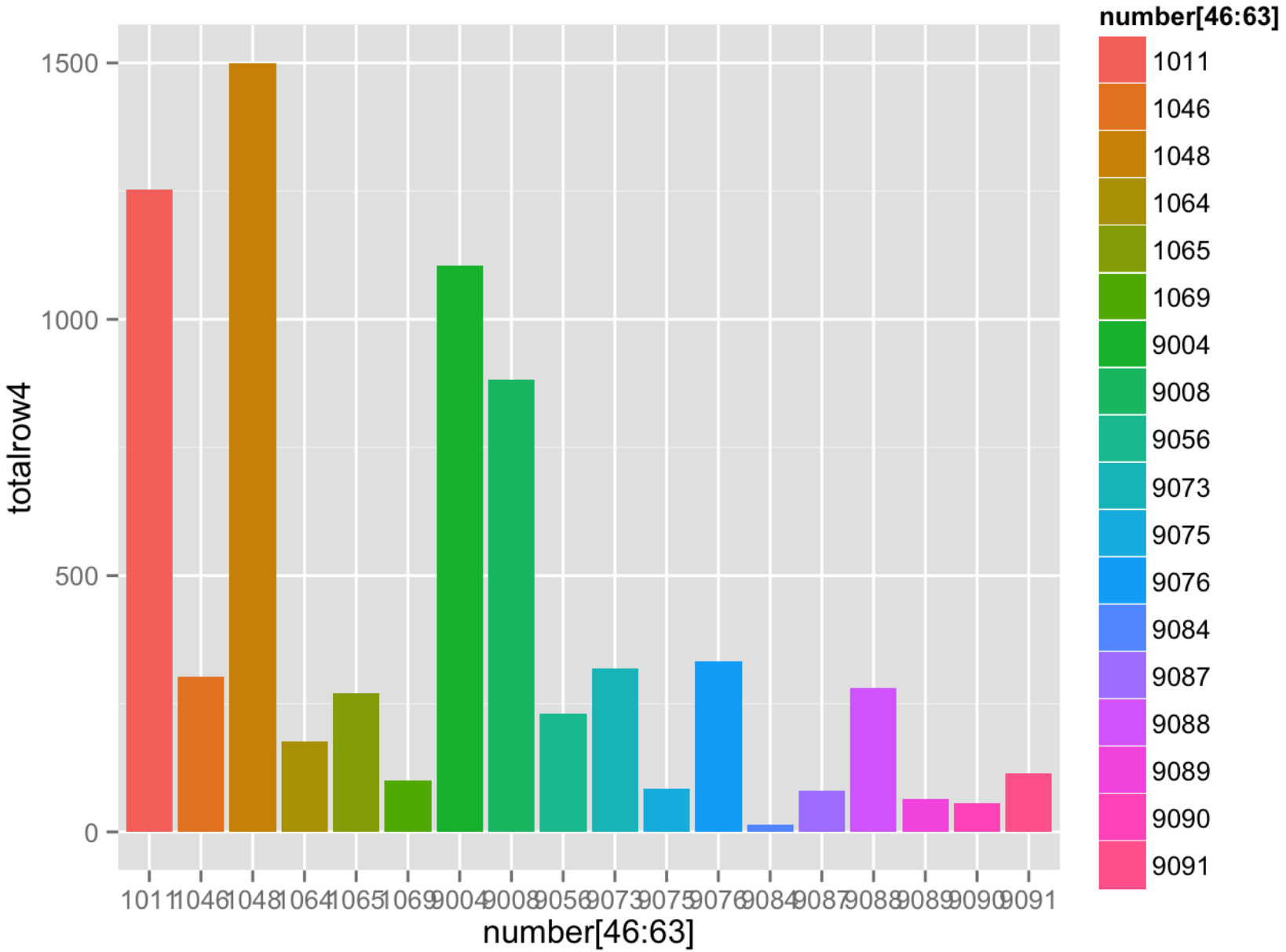


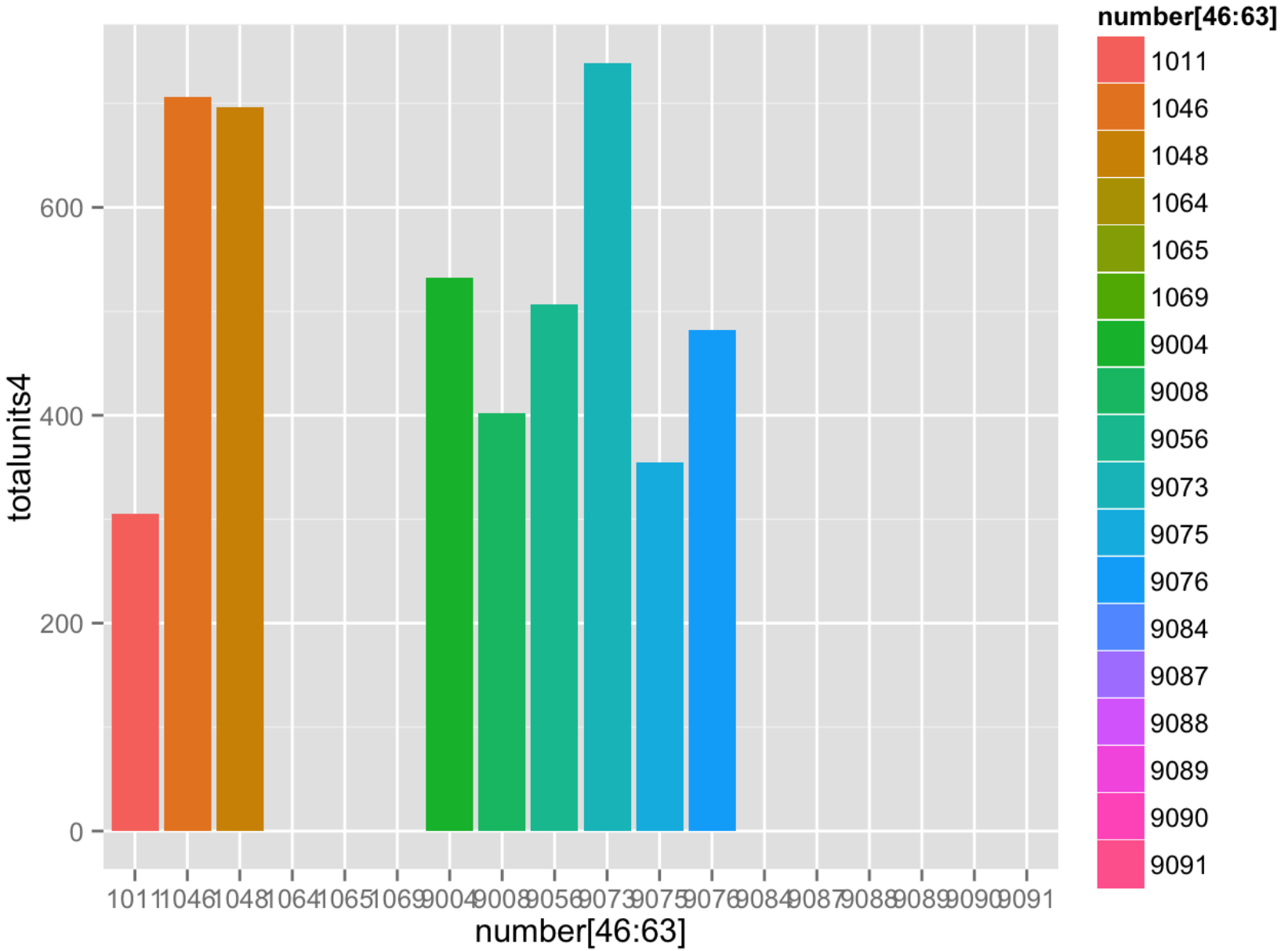


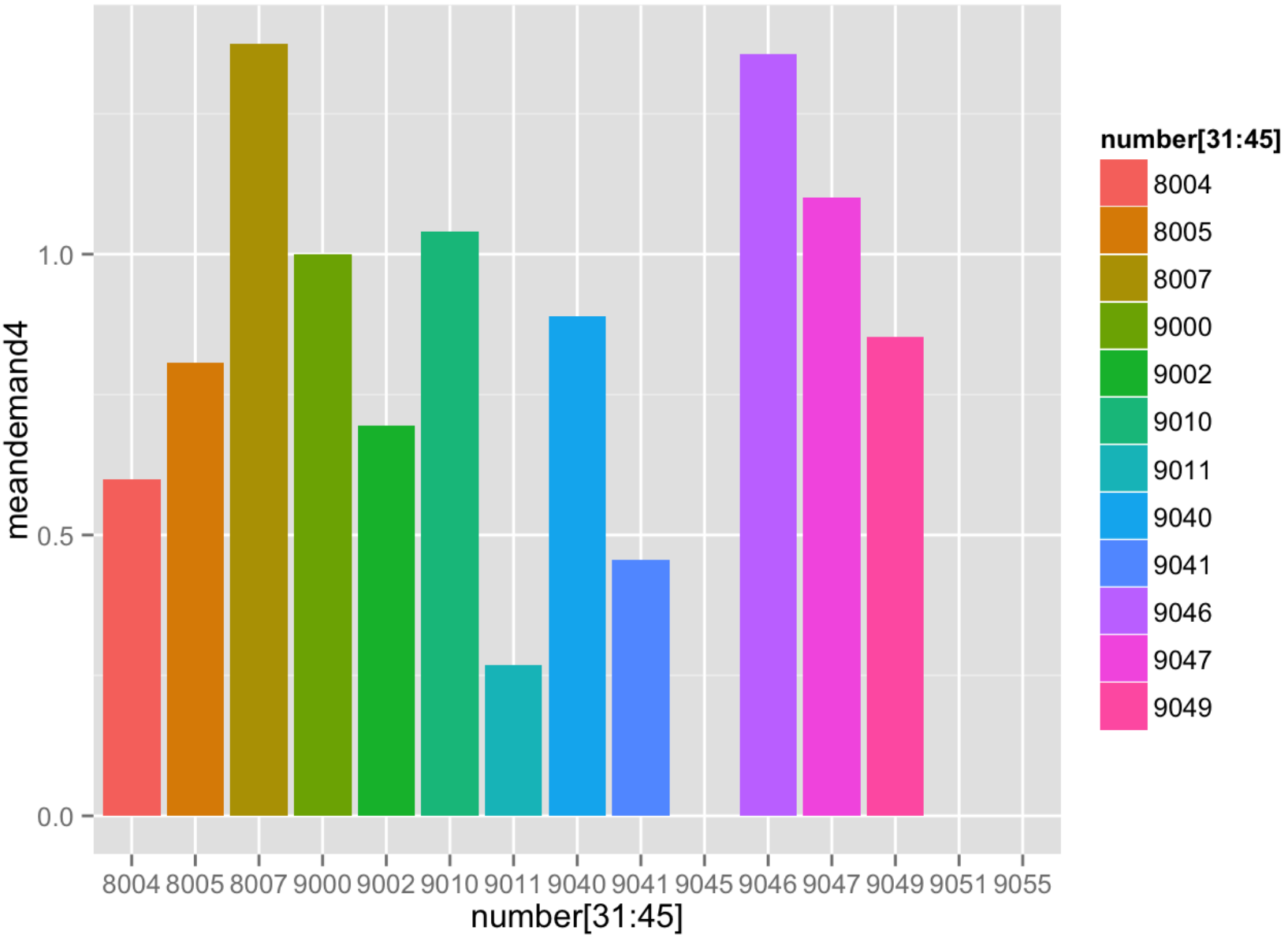













``` Now we would like to see the store which has less than 50 total units.

```
totalunits=c(totalunits1,totalunits2,totalunits3,totalunits4)
meandemands=c(meandemand1,meandemand2,meandemand3,meandemand4)
```
```

```
print(unique(number[which(totalunits<100)]))
```

```
##  [1] 1000 1030 1031 8003 9045 9051 9055 9087 9088 9089 9090 9091 1064 1065
## [15] 1069 9084
## 160 Levels: 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 ... zzz 9076
```

Now we would like to elimate these stores for analysis.

```
elimate=unique(number[which(totalunits<100)])
newdata3=newdata2[-which(newdata2$SITE.NUMBER%in%elimate),]
newdata3=newdata3[-which(newdata3$SITE.NUMBER==1015),]
newdata3=newdata3[-which(newdata3$SITE.NUMBER==1040),]
newdata3=newdata3[-which(newdata3$SITE.NUMBER==8000),]

newdata3=newdata3[which(newdata3$state=='CA'),]
newdata3$SITE.NUMBER[which(newdata3$SITE.NUMBER=='zzz 9075')]=9075
newdata3$SITE.NUMBER[which(newdata3$SITE.NUMBER=='zzz 1011')]=1011
newdata3$SITE.NUMBER[which(newdata3$SITE.NUMBER=='zzz 9076')]=9076
newdata3=newdata3[-which(newdata3$SITE.NUMBER==9075),]
newdata3=newdata3[-which(newdata3$SITE.NUMBER==9076),]
number11=unique(newdata3$SITE.NUMBER)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
stat=read.csv('mse_CA.csv')
```

```
knitr::kable(stat, format = "html")
```

X	number11	X1	X2	X3	X4	X5	meanss	totalunitss	meandemandss
1	1005	1.1720995	1.2447851	1.0878583	1.1404759	1.2046044	1.1699646	606	0.9677419
2	1006	0.7936335	0.8752805	0.8681538	0.8838754	0.9291254	0.8700137	976	0.2794118
3	1007	1.3351735	1.1688611	1.6440349	1.7777863	1.4949842	1.4841680	663	0.7500000
4	1008	1.3268043	1.8309827	1.5003903	1.5499877	1.8993503	1.6215031	799	1.5925926
5	1009	4.3507725	2.9654707	6.0624278	4.3606030	3.6838814	4.2846311	1608	2.8297872
6	1010	1.2035018	1.0251375	0.9989989	1.0758823	0.9276646	1.0462370	499	0.9555556
7	1011	1.5922496	1.5704336	1.2561042	1.1942306	1.2436910	1.3713418	305	0.8695652
8	1014	0.9819361	1.1099309	0.9547782	1.0708933	1.1331118	1.0501300	554	0.8055556
9	1016	1.8696942	2.3468179	2.8374273	2.4462260	2.2140169	2.3428364	635	3.2000000
10	1017	0.8675394	0.9545530	1.0262472	0.9992576	1.0135478	0.9722290	404	1.0500000
11	1018	1.3309749	1.1829252	1.2930920	1.1848874	1.3376800	1.2659119	402	0.6458333
12	1020	1.5975989	1.7598603	1.4355052	1.6571846	1.7455052	1.6391309	453	2.3571429
13	1021	1.3221875	1.6000762	1.6461182	1.3551245	1.3892949	1.4625603	399	0.6842105
14	1023	1.4345790	1.5243331	1.4495688	1.5806817	1.5630431	1.5104412	687	1.1000000
15	1026	1.6355930	1.2453480	1.6634106	1.7739210	1.7745792	1.6185704	886	0.7647059
16	1027	1.3000797	1.3968561	1.4300978	1.2822746	1.3983958	1.3615408	558	1.1111111

17	1028	0.9183672	0.9574954	1.1371979	0.9580545	1.0278922	0.9998014	630	0.5681818
18	1029	0.9625822	1.2663975	1.1562688	1.0907191	1.1091132	1.1170161	501	0.8000000
19	1032	0.9889998	1.1561267	1.2444253	0.9874472	1.1021306	1.0958259	613	0.7307692
20	1033	0.9034563	0.8330403	0.9199456	0.8603584	0.8368222	0.8707245	715	0.2619048
21	1034	1.3330298	1.0651863	1.2593618	1.1574329	1.2985410	1.2227104	357	1.1200000
22	1037	0.7562577	0.8751826	0.8543998	0.7402963	0.7647421	0.7981757	360	0.4150943
23	8001	1.7034733	1.6835894	1.4233727	1.5337471	1.5286287	1.5745622	678	2.3076923
24	8004	1.0341525	1.0388568	1.1092692	1.1329023	1.0248636	1.0680089	742	0.6111111
25	8005	1.1737657	1.1112367	1.2187657	1.0265424	1.0532098	1.1167040	498	1.0322581
26	8007	1.3420408	1.3105765	1.2362036	1.4863470	1.4230084	1.3596353	496	1.2916667
27	9000	1.0705122	1.1736196	1.1405008	1.2389285	1.1764940	1.1600110	1112	0.8703704
28	9002	1.1066941	1.3051638	1.1431765	1.2653384	1.4661240	1.2572993	482	0.4571429
29	9010	1.0507402	1.4248500	1.4819281	1.2977581	1.0284102	1.2567373	700	0.5961538
30	9011	1.1852462	1.1706420	0.8906400	0.9568812	0.8501245	1.0107068	560	0.2857143
31	9040	0.9720534	1.4025609	0.7212720	0.9987670	0.9904292	1.0170165	391	0.6111111
32	9046	2.0216038	1.3808556	1.6360765	1.9732178	1.6967797	1.7417067	560	1.1764706
33	9047	1.2955268	1.6744987	1.8340572	1.5666136	1.6115950	1.5964583	448	0.9333333

34	9041	1.1697533	1.1562356	0.7272816	0.9507804	0.6720897	0.9352281	398	0.2982456
35	9049	0.8640788	0.7663606	0.7740572	0.9014875	0.9100830	0.8432134	425	0.5200000
36	9008	2.2594420	1.7573098	1.6330631	2.0076180	1.5035148	1.8321895	402	0.6111111
37	9004	1.5203060	1.5436202	2.8949840	2.6241404	1.0305288	1.9227159	532	1.0322581
38	1046	2.5779957	2.5338868	2.5945360	2.3638644	2.5662168	2.5273000	706	1.2916667
39	1048	1.2373210	1.1897544	1.5930380	0.7325937	1.0002647	1.1505944	696	0.8703704
40	9056	1.7974808	2.4972843	1.1843091	1.5685426	2.3827153	1.8860664	507	0.4571429
41	9073	0.7748636	0.7683345	0.4984916	1.1392221	3.0098792	1.2381582	739	1.0322581

From the data above we found that STORE 9073 is a good choice because it has a relatively low mse ,high total units and appropriate deamand.