

Practical 4

MAKING AND USING GRAPHS IN R

Before you even begin conducting inferential statistical testing, it is always a good idea to explore your data visually and to get a sense of the structure of the data. R provides different ways to get a summary of the data, so you can look at the measures of dispersion (such as standard deviation or quartiles) and measures of central tendency (such as mean or median).

We have looked at using boxplots and histograms to get a feel for the structure of the data. In this lab, we'll look at a few more graphing options that R provides.

Simple Line Graphs

Assume that you have collected some data on the number of different drinks sold in a café. The data look like this...

	Mon	Tue	Wed	Thu	Fri
Tea	5	6	4	3	2
Coffee	8	9	5	6	6
Hot chocolate	3	4	5	4	3
Lemonade	3	1	2	2	1

We could create a table for these data directly in R...

```
>tea<-c(5,6,4,3,2)
>coffee<-c(8,9,5,6,6)
>hot_chocolate<-c(3,4,5,4,3)
>lemonade<-c(3,1,2,2,1)
```

Or we could create a text or csv file called 'beverages', and use File / Import Dataset / From Text (base)

This should give an option to import the file, and then shows the table at the top of your window.

You'll also have the following R command:

```
> beverages <- read.csv("C:/Program Files/R/beverages.csv")
> view(beverages)
```

Then type:

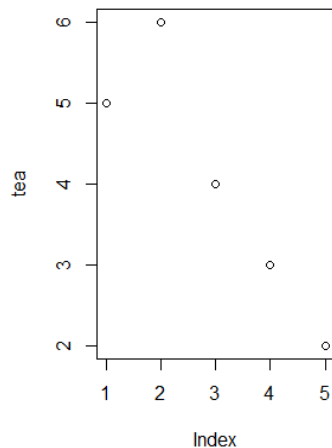
```
>attach(beverages)
```

You can make sure that the data is imported by typing:

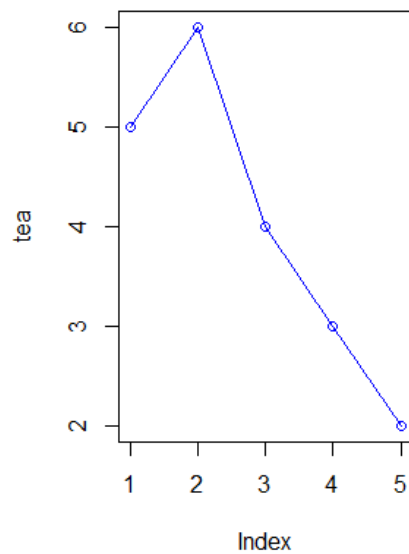
```
>show(beverages)
```

Now we'll make line graphs for these data.

```
>plot(tea)
```



We plot an overlaid line, using type "o" and assign a colour to this...

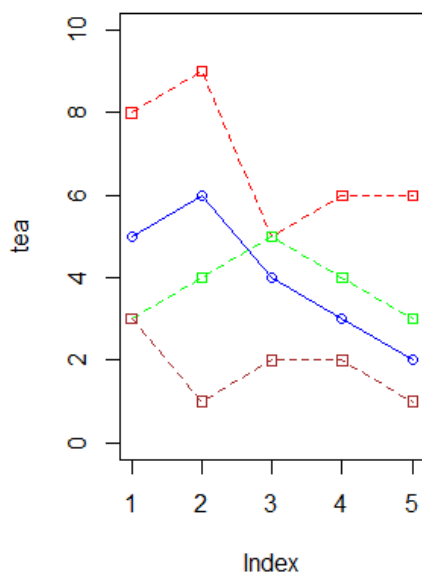


But we want to make this more useful. We add the other data for comparison...First, we need to make sure that the y-axis is large enough to include the other values. The largest number in our set is 9, so we need to have the y-axis range 0 to 10.

```
> plot(tea, type = "o", col = "blue", ylim = c(0,10))
```

We can then add the other drinks...

```
> lines(coffee,type="o",pch=22,lty=2,col="red")  
> lines(hot_chocolate,type="o",pch=22,lty=2,col="green")  
> lines(lemonade,type="o",pch=22,lty=2,col="brown")
```



Next, we want to label the x-axis with the days of the week.

First, we need remove the existing labels on the graph...

```
plot(tea, type= "o", col = "blue", axes=FALSE, ann=FALSE, ylim = c(0,10))
```

Then we add the x-axis labels that correspond to days of the week and y-axis for the full range of the set (we'll call this 'd_full'):

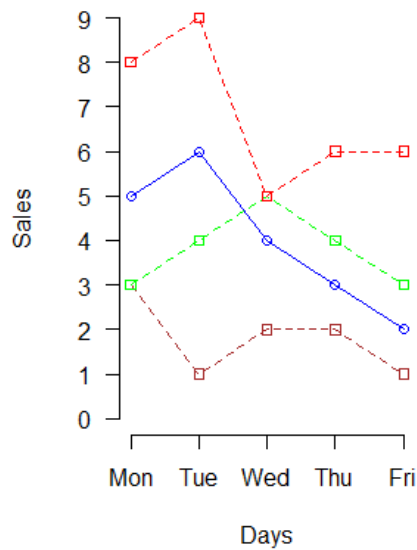
```
> d_full<-range(0,tea,coffee,hot_chocolate,lemonade)
> show(d_full)
[1] 0 9

> axis(1, at=1:5, lab=c("Mon","Tue","Wed","Thu","Fri"))
> axis(2, las=1, at=0:d_full[2])
```

And we want to provide labels for the axes...

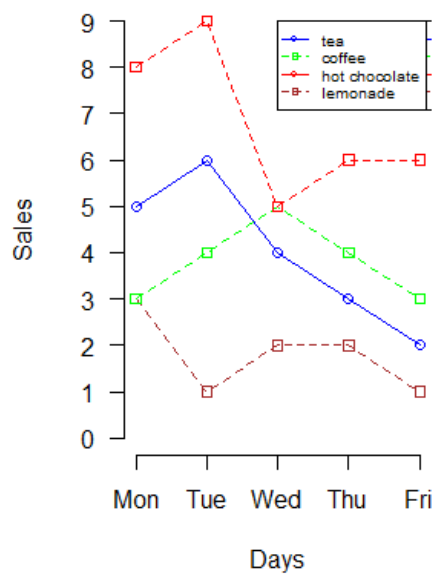
```
> title(xlab="Days", col.lab = "black", font.main=4)
> title(ylab="Sales", col.lab="black", font.main=4)
```

Then we can add back the other lines...



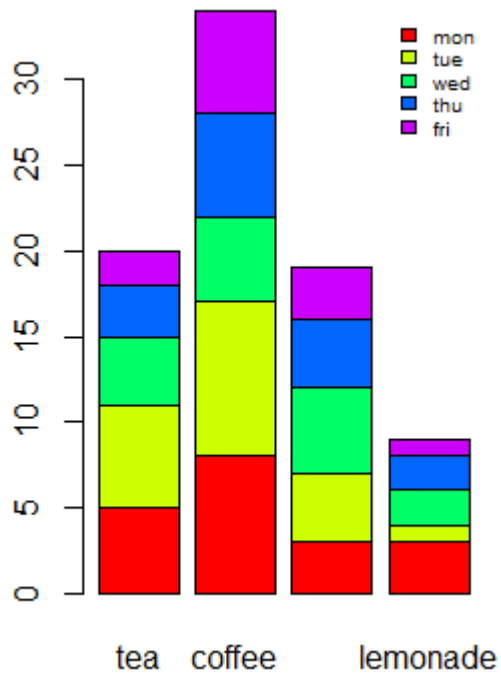
Finally, we want to add the Legend to explain what the different colours mean...

```
> legend(3,d_full[2],c("tea", "coffee", "hot chocolate", "lemonade"), cex=0.8,col=c("blue", "green","red", "brown"), pch=1:2,lty=1:2)
```



Stacked Barchart

```
> barplot(as.matrix(bev_data), col=rainbow(5))
> legend("topright", c("mon", "tue", "wed", "thu", "fri"), cex=0.6, bty="n", fill=rainbow(5))
```



Pie Chart

```
> pie(coffee, main="coffee sales", col=rainbow(length(coffee)), labels = c("mon", "tue", "wed", "thu", "fri"))
```

coffee sales



In this practical, we have provided a couple of simple exercises to get you started.

If these are too easy for you, have a look at:

<https://www.r-graph-gallery.com/all-graphs/>

A couple of examples of graphs from this site are given below.

There are also some excellent resources and tools under the ggplot function that R supports. This tutorial is pretty comprehensive:

<http://r-statistics.co/Complete-Ggplot2-Tutorial-Part1-With-R-Code.html>

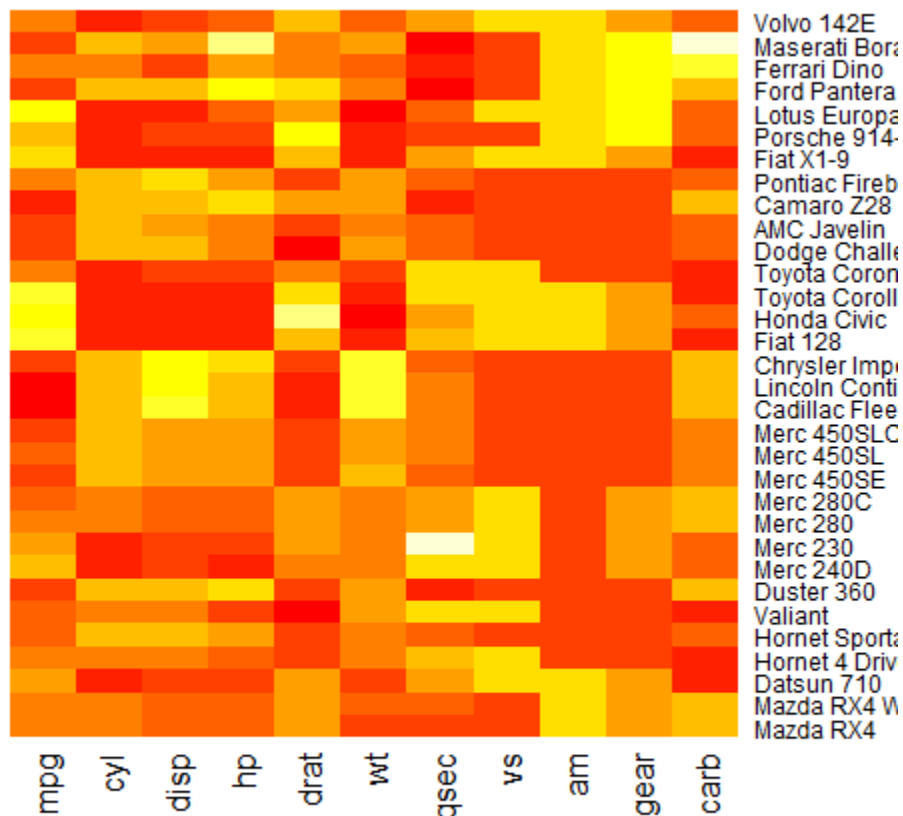
Heatmap

Using the Mtcars dataset that we looked at last week, we can create variations on a heatmap.

```
>mtcars
```

```
>data=as.matrix(mtcars)
```

```
>heatmap(data, Colv = NA, Rowv = NA, scale="column")
```



you can play with the colours...

```
> heatmap(data, Colv = NA, Rowv= NA, scale="column", col = terrain.colors(256))
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105
106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126
127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147
148	149	150	151	152																
153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173
174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194
195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215
216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236
237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257
258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278
279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299
300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341
342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362
363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383
384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404
405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425
426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446
447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467
468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488
489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509
510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530
531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551
552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572
573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593
594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614
615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635
636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656