

SPACE

The Final Frontier

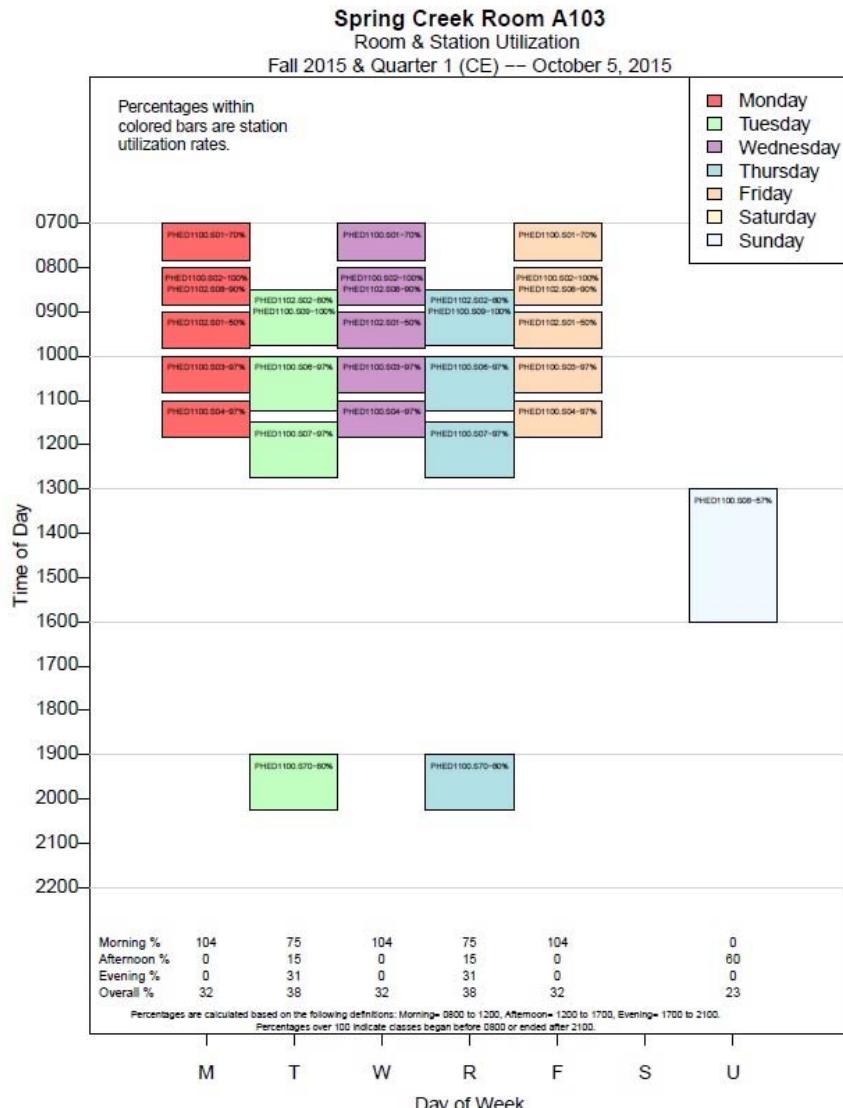
Visual Representation of Classroom Usage



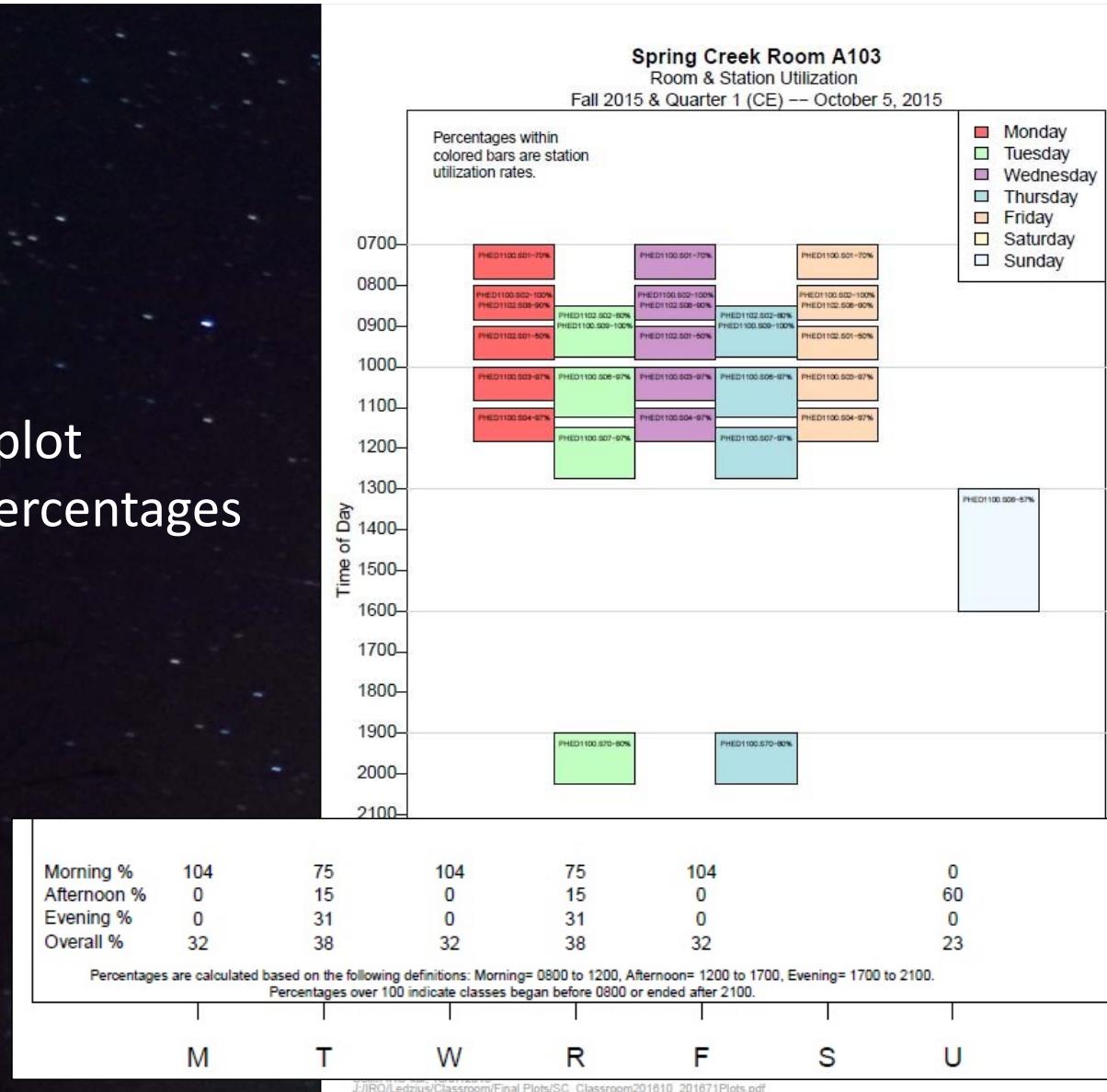
Part One: Visual Report of Classroom Usage



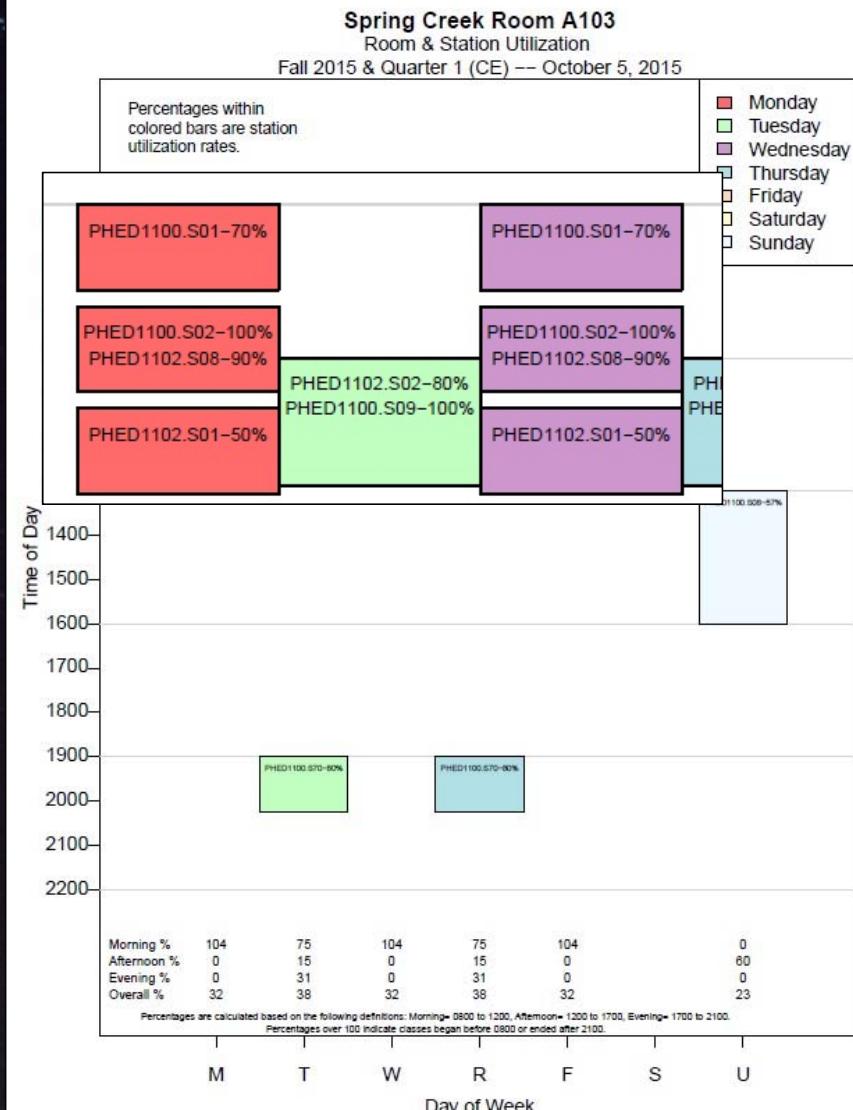
- One Page per Classroom
- Horizontal Axis – Day of Week
- Vertical Axis – Time of Day
- Color Blocks represent class time
- Report reflects usage for classes



- Percentages beneath plot
- Definitions beneath percentages



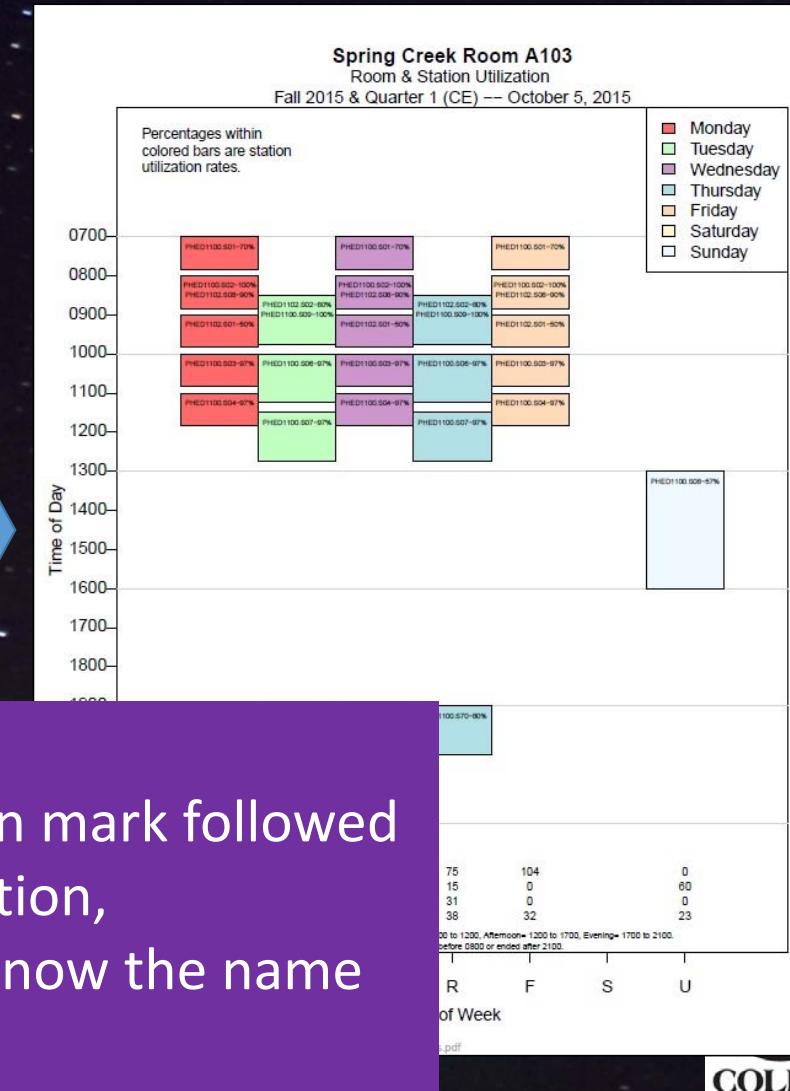
➤ Station Utilization Rates in Blocks



Collin IROkal; 10/07/2015
J:\IRO\Ledzizus\Classroom\Final Plots\SC_Classroom201610_201671Plots.pdf

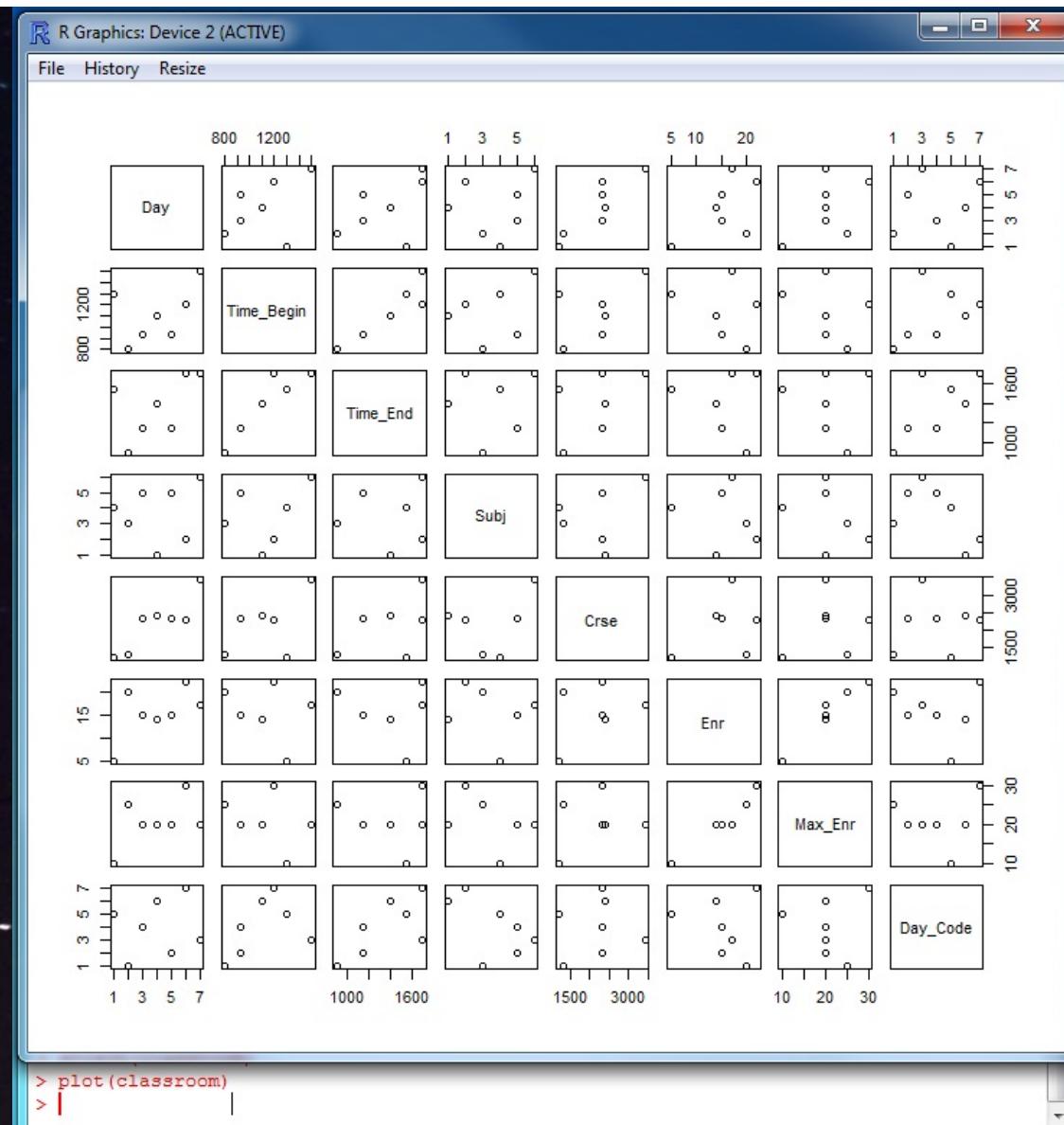
Term	Campus	DayEvening	Day	Ct	T_0700	T_0800	CRN	Time_Begin	Time_End	Bldg	Room	Subj	Crse	Sec	Enr	Max_Enr	Room_Code	MultipleUse	Mucode	Mucode1	Mucode2
201610	SC	Day	F	1			10238	900	950	SC	A100	PHED	1104	S01	20	30	1	FALSE	0	1	1
201610	SC	Day	F	1			10294	1000	1050	SC	A100	PHED	1117	S01	16	30	1	FALSE	0	1	1
201610	SC	Day	M	1			10238	900	950	SC	A100	PHED	1104	S01	20	30	1	FALSE	0	1	1
201610	SC	Day	M	1			10294	1000	1050	SC	A100	PHED	1117	S01	16	30	1	FALSE	0	1	1
201610	SC	Day	M	1			50094	1200	1315	SC	A100	PHED	1115	S02	19	20	1	FALSE	0	1	1
201610	SC	Day	M	1			10306	1300	1415	SC	A100	PHED	1123	S01	16	30	1	FALSE	0	1	1
201610	SC	Day	M	1			10308	1430	1545	SC	A100	PHED	1125	S01	24	30	1	FALSE	0	1	1
201610	SC	Evening	M	1			10347	1730	1845	SC	A100	PHED	1117	S70	15	20	1	FALSE	0	1	1
201610	SC	Evening	M	1			10348	1730	1845	SC	A100	PHED	1118	S70	6	10	1	TRUE	1	2	2
201610	SC	Day	R	1		800	10295	830	945	SC	A100	PHED	1117	S02	16	30	1	FALSE	0	1	1
201610	SC	Day	R	1			10251	1000	1115	SC	A100	PHED	1112	S01	29	30	1	FALSE	0	1	1
201610	SC	Day	R	1			10248	1130	1245	SC	A100	PHED	1111	S01	21	30	1	FALSE	0	1	1
201610	SC	Day	R	1			10292	1130	1245	SC	A100	PHED	1115	S01	20	20	1	TRUE	1	2	2
201610	SC	Day	R	1			10253	1300	1415	SC	A100	PHED	1114	S01	16	32	1	FALSE	0	1	1
201610	SC	Day	R	1			10299	1300	1415	SC	A100	PHED	1118	S01	17	30	1	TRUE	1	2	2
201610	SC	Day	R	1			19283	1600	1715	SC	A100	PHED	1142	S01	23	30	1	FALSE	0	1	1
201610	SC	Day	R	1			19284	1600	1715	SC	A100	PHED	2142	S01	20	30	1	TRUE	1	2	2
201610	SC	Evening	R	1			10346	1900	2145	SC	A100	PHED	1114	S70	15	30	1	FALSE	0	1	1
201610	SC	Day	T	1		800	10295	830	945	SC	A100	PHED	1117	S02	16	30	1	FALSE	0	1	1
201610	SC	Day	T	1			10251	1000	1115	SC	A100	PHED	1112	S01	29	30	1	FALSE	0	1	1
201610	SC	Day	T	1			10248	1130	1245	SC	A100	PHED	1111	S01	21	30	1	FALSE	0	1	1
201610	SC	Day	T	1			10292	1130	1245	SC	A100	PHED	1115	S01	20	20	1	TRUE	1	2	2
201610	SC	Day	T	1			10253	1300	1415	SC	A100	PHED	1114	S01	16	32	1	FALSE	0	1	1
201610	SC	Day	T	1			10299	1300	1415	SC	A100	PHED	1118	S01	17	30	1	TRUE	1	2	2
201610	SC	Day	T	1			19283	1600	1715	SC	A100	PHED	1142	S01	23	30	1	FALSE	0	1	1
201610	SC	Day	T	1			19284	1600	1715	SC	A100	PHED	2142	S01	20	30	1	TRUE	1	2	2
201610	SC	Day	T	1			11226	1900	2145	SC	A100	PHED	1111	S70	21	30	1	FALSE	0	1	1
201610	SC	Evening	T	1			10238	900	950	SC	A100	PHED	1104	S01	20	30	1	FALSE	0	1	1
201610	SC	Day	W	1			10294	1000	1050	SC	A100	PHED	1117	S01	16	30	1	FALSE	0	1	1
201610	SC	Day	W	1			50094	1200	1315	SC	A100	PHED	1115	S02	19	20	1	FALSE	0	1	1
201610	SC	Day	W	1			10306	1300	1415	SC	A100	PHED	1123	S01	16	30	1	FALSE	0	1	1
201610	SC	Day	W	1			10308	1430	1545	SC	A100	PHED	1125	S01	24	30	1	FALSE	0	1	1
201610	SC	Evening	W	1			10347	1730	1845	SC	A100	PHED	1117	S70	15	20	1	FALSE	0	1	1
201610	SC	Evening	W	1			10348	1730	1845	SC	A100	PHED	1118	S70	6	10	1	TRUE	1	2	2
201610	SC	Day	F	1	700		10196	700	750	SC	A103	PHED	1100	S01	14	20	2	FALSE	0	1	1

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Term	Campus	Day	CRN	Time_Begin	Time_End	Bldg	Room	Subj	Crse	Sec	Enr	Max_Enr
2	201610	SC	F	10238	900	950	SC	A100	PHED	1104	S01	20	30
3	201610	SC	F	10294	1000	1050	SC	A100	PHED	1117	S01	16	30
4	201610	SC	M	10238	900	950	SC	A100	PHED	1104	S01	20	30
5	201610	SC	M	10294	1000	1050	SC	A100	PHED	1117	S01	16	30
6	201610	SC	M	50094	1200	1315	SC	A100	PHED	1115	S02	19	20
7	201610	SC	M	10306	1300	1415	SC	A100	PHED	1123	S01	16	30
8	201610	SC	M	10308	1430	1545	SC	A100	PHED	1125	S01	24	30
9	201610	SC	M	10347	1730	1845	SC	A100	PHED	1117	S70	15	20
10	201610	SC	M	10348	1730	1845	SC	A100	PHED	1118	S70	6	10
11	201610	SC	R	10295	830	945	SC	A100	PHED	1117	S02	16	30
12	201610	SC	R	10251	1000	1115	SC	A100	PHED	1112	S01	29	30
13	201610	SC	R	10248	1130	1245	SC	A100	PHED	1111	S01	21	30
14	201610	SC	R	10292	1130	1245	SC	A100	PHED	1115	S01	20	20
15	201610	SC	R	10253	1300	1415	SC	A100	PHED	1114	S01	16	32
16	201610	SC	R	10299	1300	1415	SC	A100	PHED	1118	S01	17	30
17	201610	SC	R	19283	1600	1715	SC	A100	PHED	1142	S01	23	30
18	201610	SC	R	19284	1600	1715	SC	A100	PHED	2142	S01	20	30
19	201610	SC	R	10346	1900	2145	SC	A100	PHED	1114	S70	15	30
20	201610	SC	T	10295	830	945	SC	A100	PHED	1117	S02	16	30
21	201610	SC	T	10251	1000	1115	SC	A100	PHED	1112	S01	29	30
22	201610	SC	T	10248	1130	1245	SC	A100	PHED	1111	S01	21	30
23	201610	SC	T	10292	1130	1245	SC	A100	PHED	1115	S01	20	20
24	201610	SC	T	10253	1300	1415	SC	A100	PHED	1114	S01	16	32
25	201610	SC	T	10299	1300	1415	SC	A100	PHED	1118	S01	17	30
26	201610	SC	T	19283	1600	1715	SC	A100	PHED	1142	S01	23	30
27	201610	SC	T	19284	1600	1715	SC	A100	PHED	2142	S01	20	30
28	201610	SC	T	11226	1900	2145	SC	A100	PHED	1114	S70	15	30
29	201610	SC	W	10238	900	950	SC	A100	PHED	1117	S02	16	30
30	201610	SC	W	10294	1000	1050	SC	A100	PHED	1112	S01	29	30
31	201610	SC	W	50094	1200	1315	SC	A100	PHED	1111	S01	21	30
32	201610	SC	W	10306	1300	1415	SC	A100	PHED	1123	S01	16	30
33	201610	SC	W	10308	1430	1545	SC	A100	PHED	1125	S01	24	30
34	201610	SC	W	10347	1730	1845	SC	A100	PHED	1117	S01	15	20
35	201610	SC	W	10348	1730	1845	SC	A100	PHED	1118	S01	17	30
36	201610	SC	F	10196	700	750	SC	A100	PHED	1142	S01	23	30



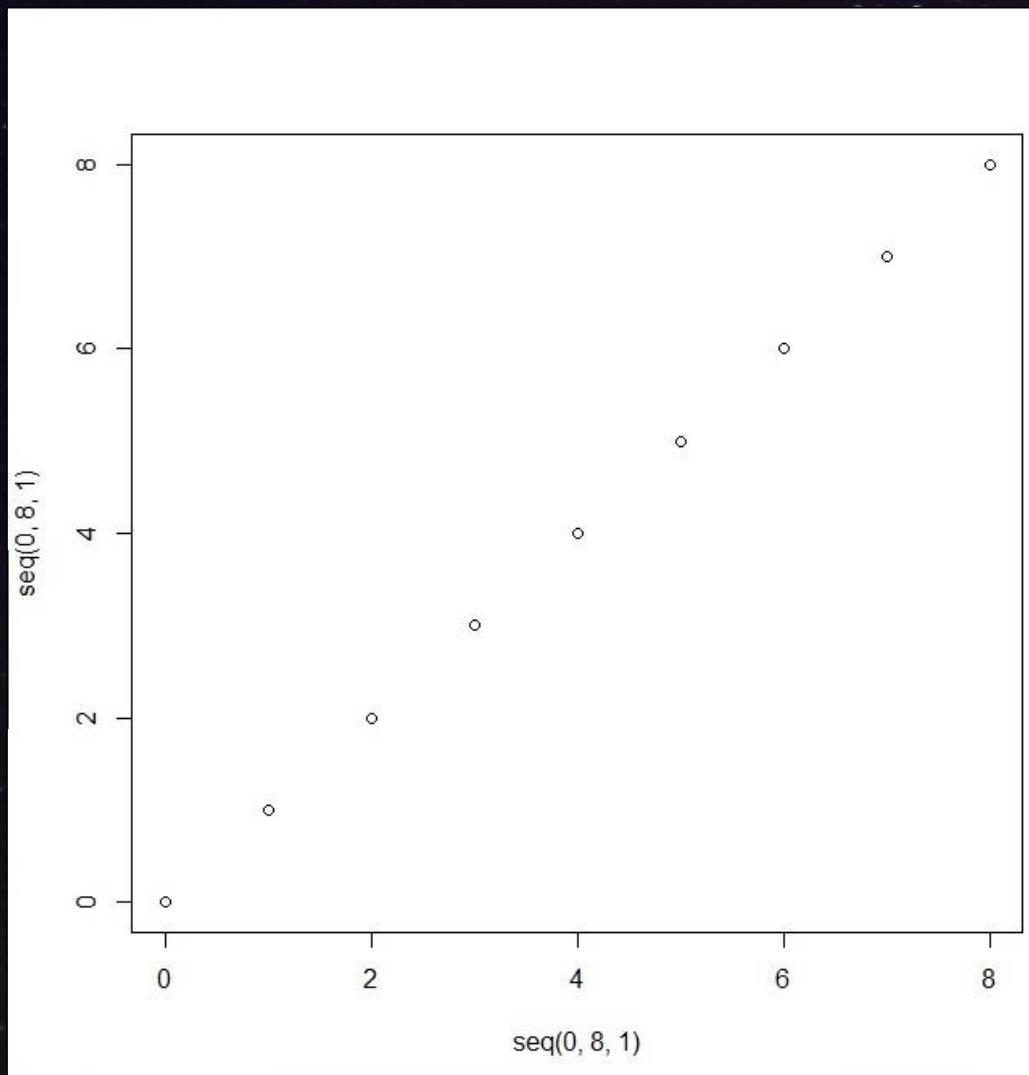
- Example of plotting entire data set.

plot(classroom)



➤ Plotting x and y.

```
plot(seq(0,8,1),seq(0,8,1))
```



- Adding more arguments to the plot function.

```
plot(seq(0,8,1),seq(0,8,1),xaxt="n",yaxt="n",
main="Main Campus",cex.main=1.2,xlab="Day of Week",
ylab="Time of Day",type="n")
```

Main Campus

Time of Day

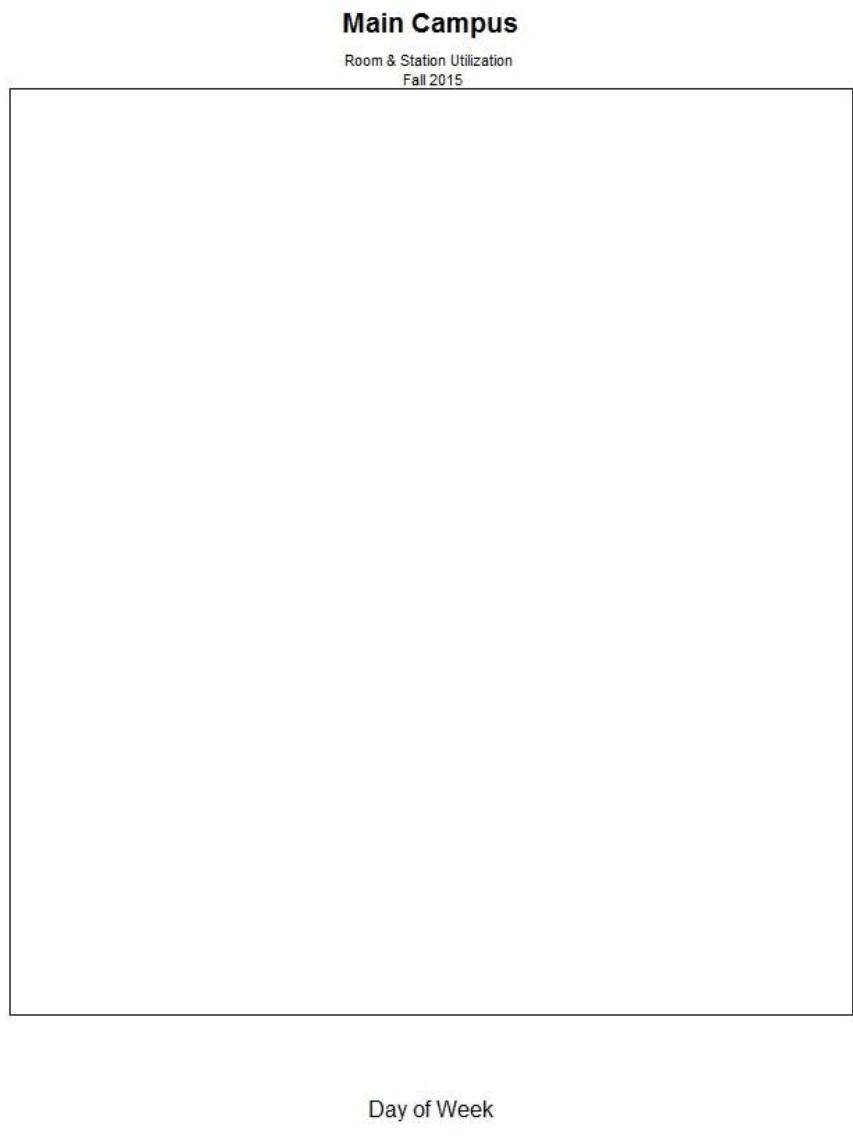
Day of Week



- Adding margin text.

```
mtext("Room & Station Utilization  
\n Fall 2015",cex=.7)
```

cex= character expansion (or font size), default is one.

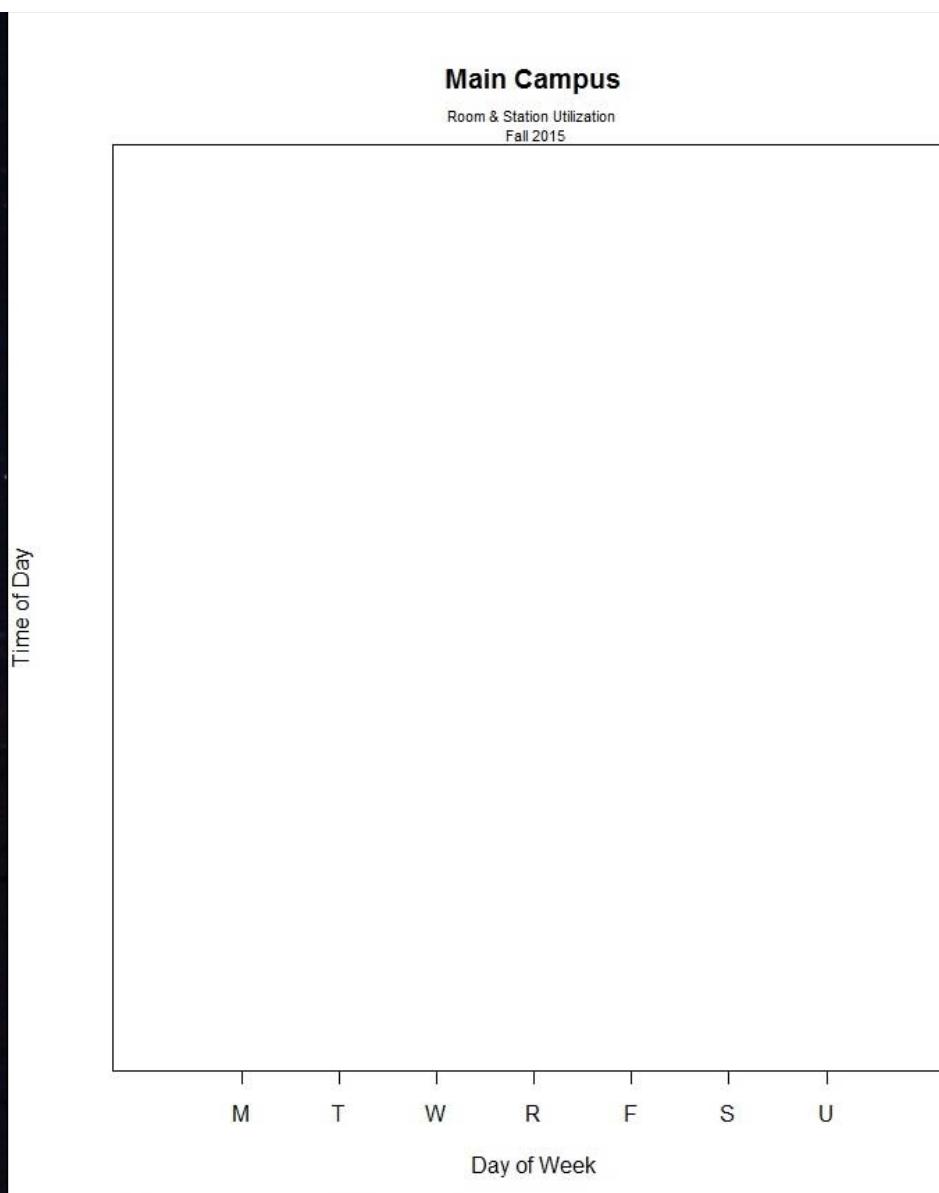


- Create a vector containing the desired axis labels.

```
lablist.x <- as.vector(c('M','T','W','R','F','S','U'))
```

- The arrow (<-)assigns the formula on the right to the name on the left.
- Place tick marks and add labels along the x-axis.

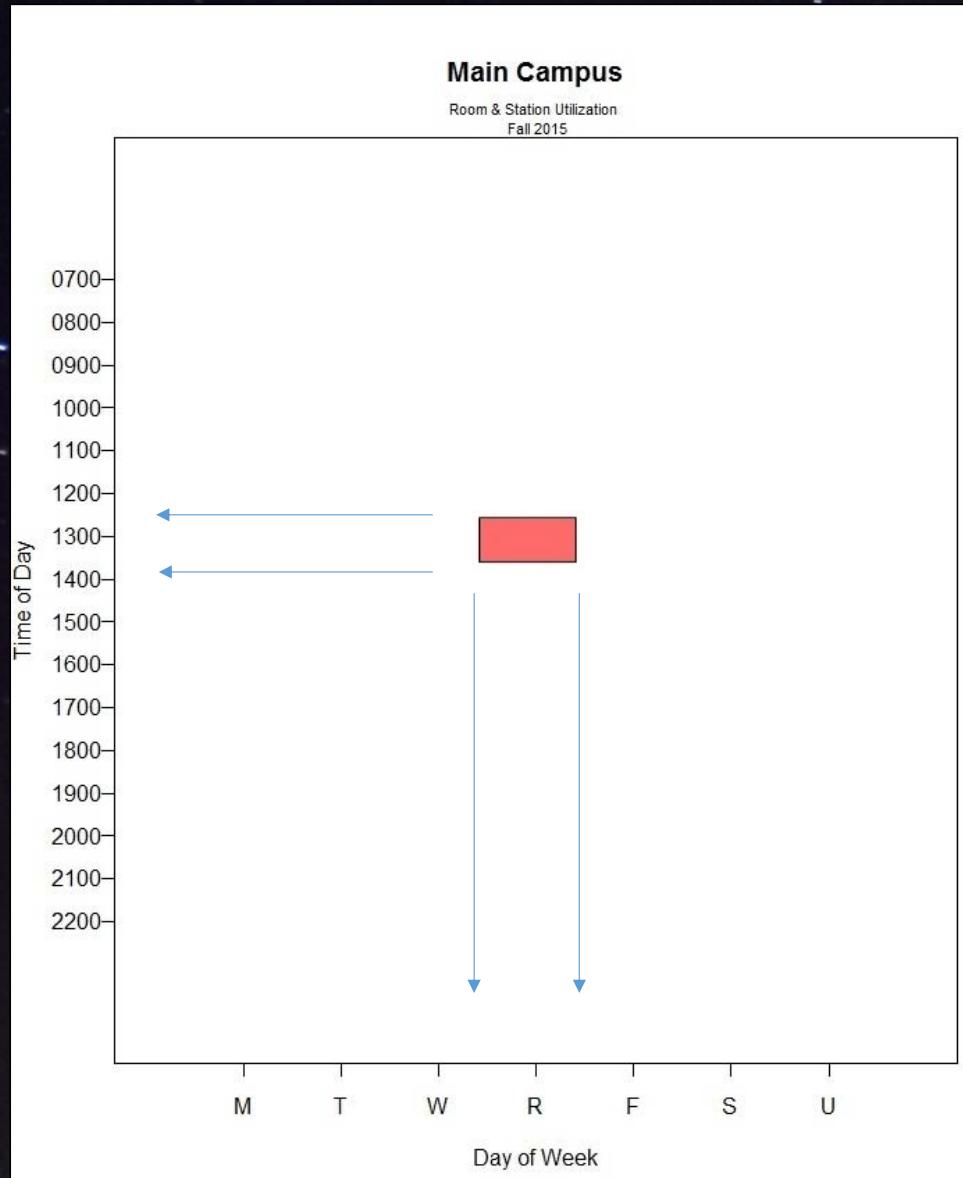
```
axis(1, at=c(1,2,3,4,5,6,7),labels=lablist.x)
```



- Do similar for y-axis.

```
lablist.y <-  
as.vector(c('2200','2100','2000','1900','1800','1700','1600','1500',  
'1400','1300','1200','1100','1000','0900','0800','0700'))  
  
axis(2, at=seq(1,7, by=.4), labels = lablist.y, las=2)
```

- In using the rectangle function, we provide four points from which R calculates the vertices of each rectangle. Two points are referenced from the x-axis and two from the y-axis.



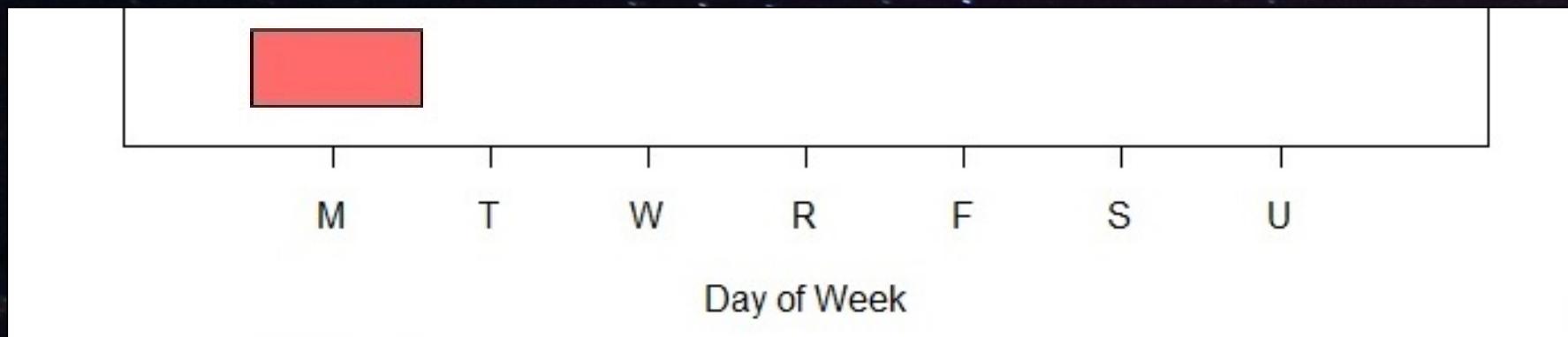
Sample data.csv - Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Term	Campus	Day	CRN	Time_Begin	Time_End	Room	Subj	Crse	Sec	Enr	Max_Enr	Room_Cod	MultipleUs	Mucode	Mucode1	Mucode2		
2	201610	SC	F	10238	900	950	A100	PHED	1104	S01	20	30	1	FALSE	0	1	1		
3	201610	SC	F	10294	1000	1050	A100	PHED	1117	S01	16	30	1	FALSE	0	1	1		
4	201610	SC	M	10238	900	950	A100	PHED	1104	S01	20	30	1	FALSE	0	1	1		
5	201610	SC	M	10294	1000	1050	A100	PHED	1117	S01	16	30	1	FALSE	0	1	1		
6	201610	SC	M	50094	1200	1315	A100	PHED	1115	S02	19	20	1	FALSE	0	1	1		
7	201610	SC	F	10196	700	750	A103	PHED	1100	S01	14	20	2	FALSE	0	1	1		
8	201610	SC	F	10200	800	850	A103	PHED	1100	S02	20	20	2	FALSE	0	1	1		
9	201610	SC	F	50140	800	850	A103	PHED	1102	S08	9	10	2	TRUE	1	2	2		
10	201610	SC	F	10230	900	950	A103	PHED	1102	S01	15	30	2	FALSE	0	1	1		
11	201610	SC	F	10207	1000	1050	A103	PHED	1100	S03	29	30	2	FALSE	0	1	1		
12	201610	SC	F	10334	800	850	A107	PHED	1147	S01	15	25	3	FALSE	0	1	1		
13	201610	SC	F	10314	900	950	A107	PHED	1126	S01	21	25	3	FALSE	0	1	1		

- Recode character to numeric for calculations.

Day_Code <-

```
ifelse(Day=="F",5,ifelse(Day=="M",1,ifelse(Day=="T",2,ifelse(Day=="W",3,ifelse(Day=="R",4,ifelse(Day=="S",6,ifelse(Day=="U",7,0)))))))
```



- Assigning values to Point1 and Point2.

Point1 <- Day_Code-.5

Point2 <- Day_Code+.5

- Calculate the vertices of the rectangle that correspond to y-axis.

```
substrRight <- function(x, n){ substr(x, nchar(x)-n+1, nchar(x))}  
substrLeft <- function(x, n){ substr(x, 1, nchar(x)-2)}
```

```
StartHr <- as.numeric(substrLeft(Time_Begin,2))  
StartMin <- as.numeric(substrRight(Time_Begin,2))  
Point3 <- (((StartHr-7)+(StartMin/60))*(-0.4))+7
```

```
EndHr <- as.numeric(substrLeft(Time_End,2))  
EndMin <- as.numeric(substrRight(Time_End,2))  
Point4 <- (((EndHr-7)+(EndMin/60))*(-0.4))+7
```

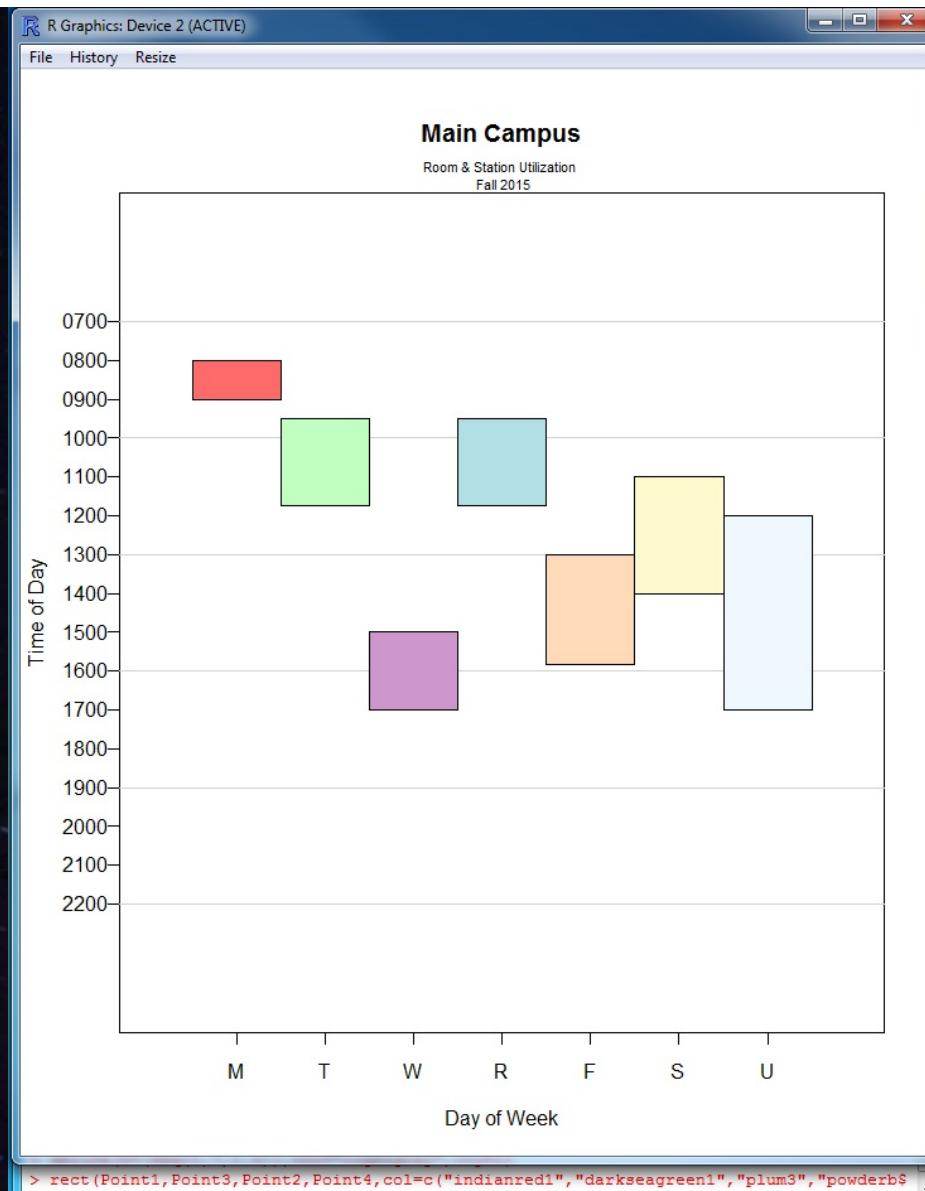
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	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	657																		

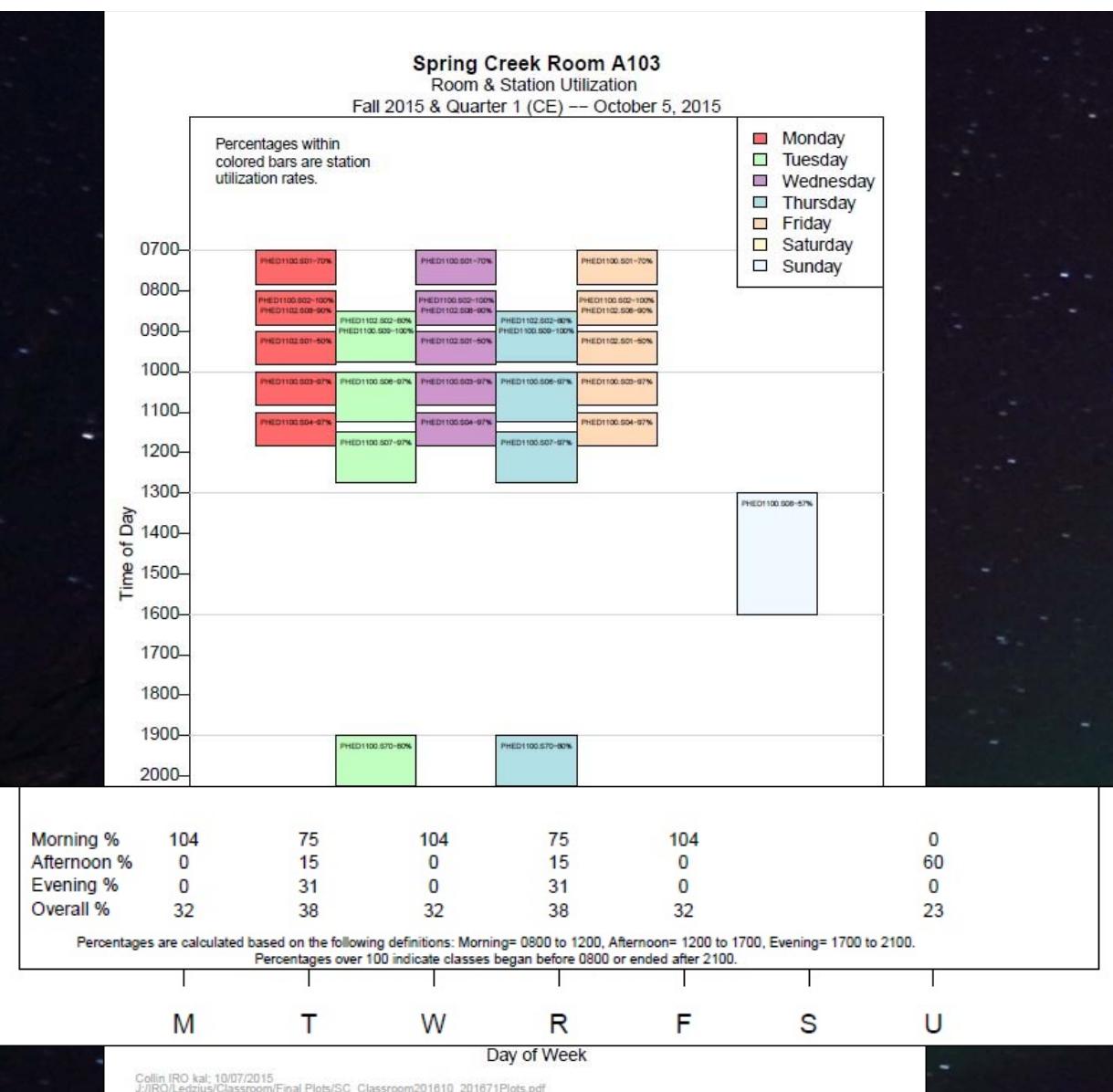
➤ In R, type in ‘colors()’ to get a list of the hundreds of color names.

- Draw rectangles representing blocks of time.

```
rect (xleft, ybottom, xright, ytop., ...)
```

```
rect(Point1,Point4,Point2,Point3,  
col=c("indianred1","darkseagreen1","plum3","powderblue",  
"peachpuff","lemonchiffon","aliceblue")[Day_Code],border="black")
```





```
BegTimeCode <- (StartHr*100+(StartMin/60)*100)
EndTimeCode <- (EndHr*100+(EndMin/60)*100)
```

➤ Define morning, afternoon, evening, and overall for R.

```
MorningHrs<-ifelse(Time_Begin<1200 & Time_End<=1200,
(EndTimeCode-BegTimeCode)/4,ifelse(Time_Begin<1200 &
Time_End>1200,(1200-BegTimeCode)/4,0))
```

```
AfternoonHrs<-ifelse(Time_Begin>=1200 & Time_Begin<1700 &
Time_End>1200 & Time_End<=1700,(EndTimeCode-BegTimeCode)/5,
ifelse(Time_Begin>=1200 & Time_Begin<1700 & Time_End>1700,
(1700-BegTimeCode)/5,ifelse(Time_Begin<1200 & Time_End>1200 &
Time_End<=1700,(EndTimeCode-1200)/5,
ifelse(Time_Begin<1200 & Time_End>1700,100,0))))
```

```
EveningHrs<-ifelse(Time_Begin>=1700 & Time_End>1700,
(EndTimeCode-BegTimeCode)/4,ifelse(Time_Begin<=1700 &
Time_End>1700,(EndTimeCode-1700)/4,0))
```

```
Overall <-(MorningHrs*4+AfternoonHrs*5+EveningHrs*4)/13
```

- Count each space only once even if there are multiple classes meeting in the same space at the same time. Create a variable to use in unduplicating same-time room usage.

```
undup <- paste(Room_Code,Day_Code,StartHr)
```

-
- Use created variable to subset morning hours so that classrooms with multiple usage are only counted once.

```
UndupMorn <- as.numeric(subset(MorningHrs,!duplicated(undup)))
```

- Concatenate room and day.

```
t <- paste(Room_Code, Day_Code)
```

- Create a vector of all room/day combinations without repeating any.

```
tog <- subset(t, !duplicated(undup))
```

- Use grouping variable in summing the unduplicated morning hours, grouping by room and day.

```
CumM <- by(UndupMorn, tog, sum)
```

- Create variable to indicate whether the calculations are from the same day or a new day.

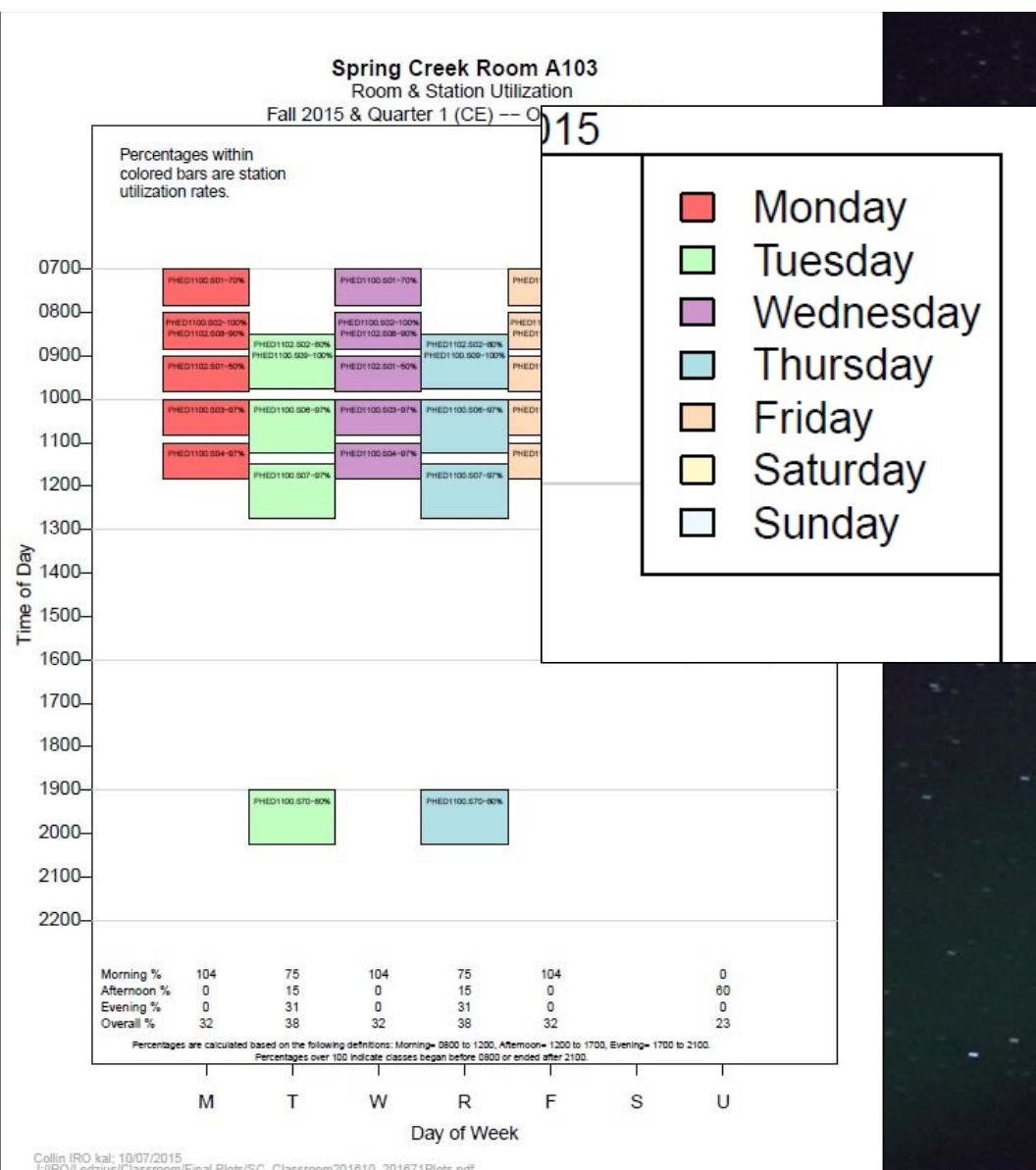
```
Indicator <- ifelse(duplicated(t)==TRUE,0,1)
```

- Create a variable that checks for new room, new day, and rounds or truncates for your specifications.

```
finalm <- ifelse(Indicator==1,floor(.5+CumM[t])," ")))
```

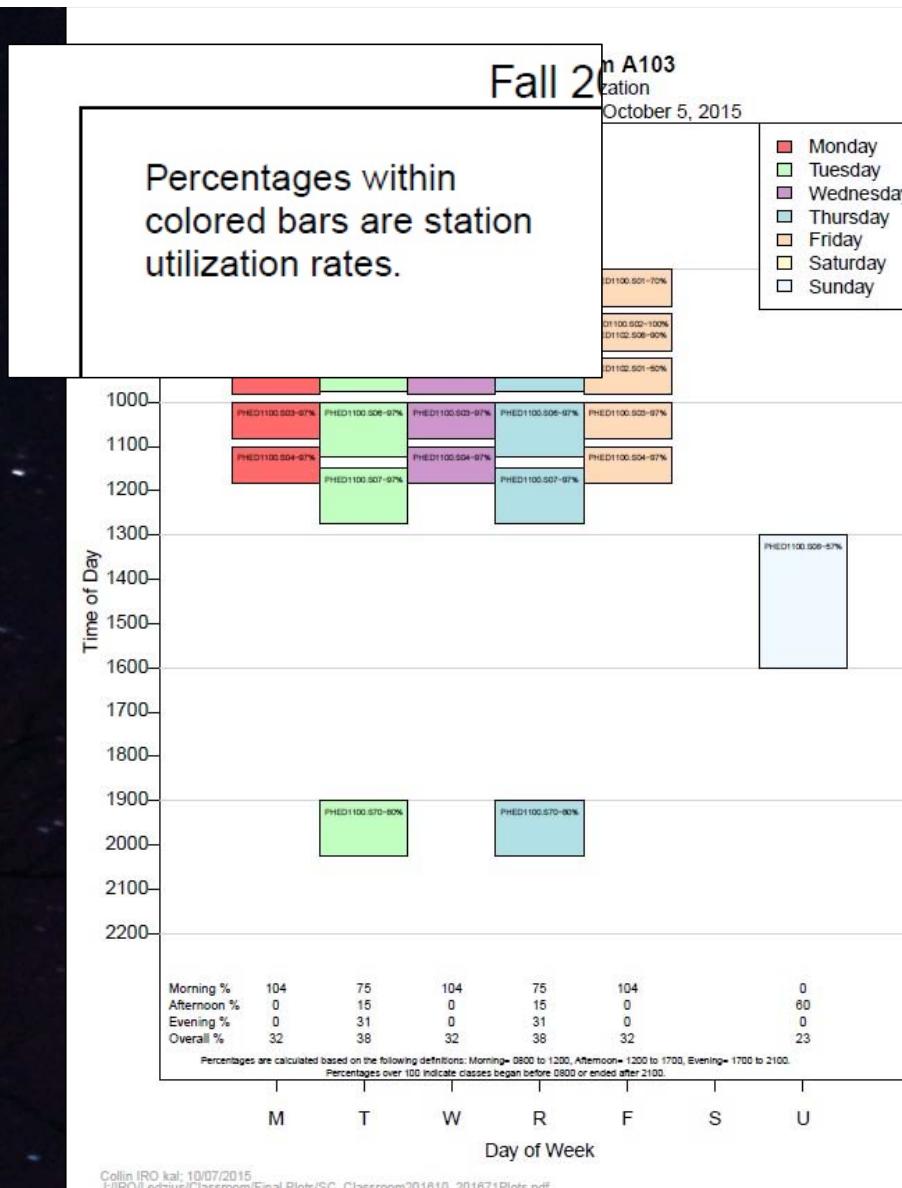
- Use text function to print calculations.

```
text(Point1+.5,.5,finalm,cex=.7)
```



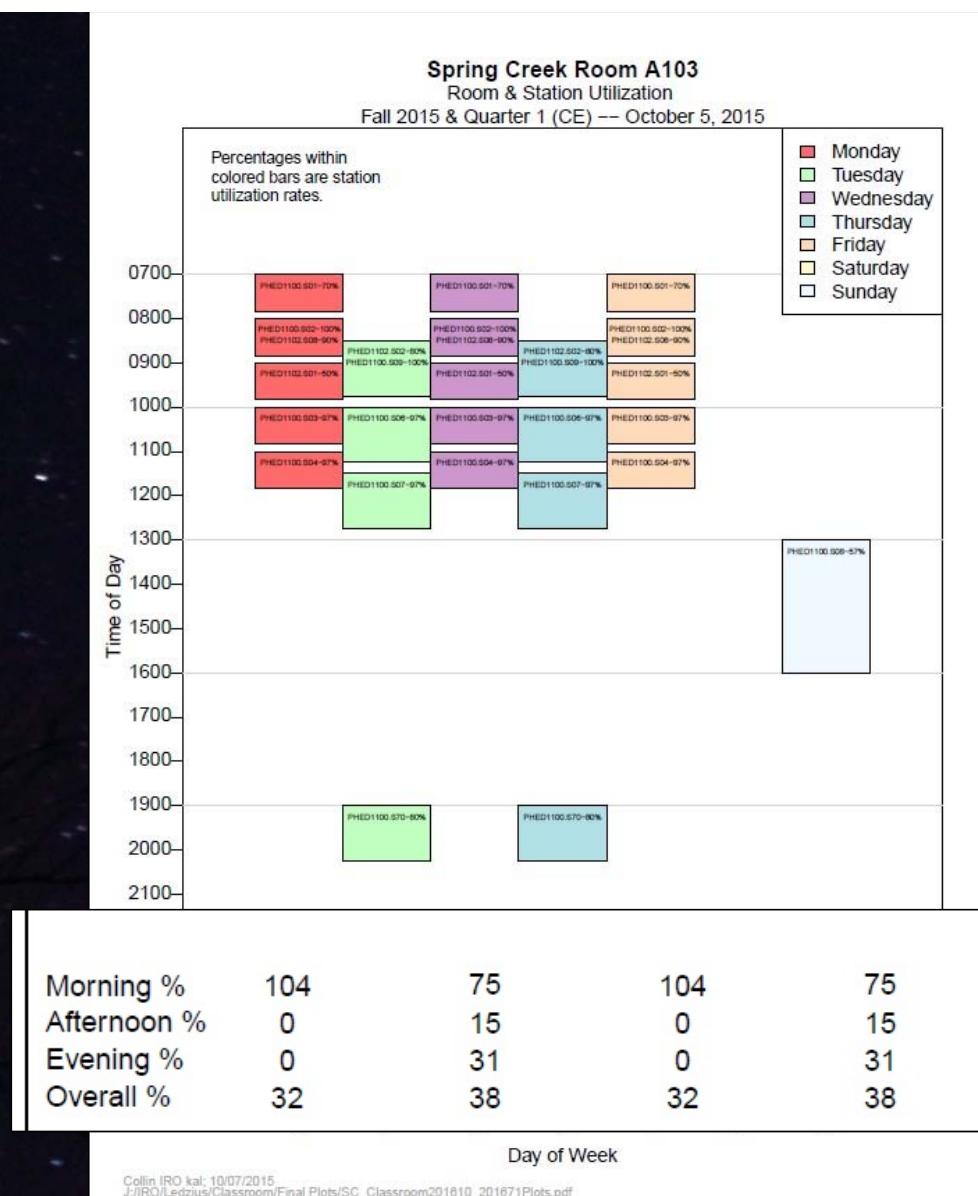
- Add a legend to the plot.

```
legend("topright",c("Monday","Tuesday","Wednesday","Thursday",
"Friday","Saturday","Sunday"),cex=1,fill=c("indianred1",
"darkseagreen1","plum3","powderblue","peachpuff",
"lemonchiffon","aliceblue"),bg="white")
```



- Other uses of legend function.

```
legend("topleft",c("Percentages within \ncolored bars are station\n\\nutilization rates."),cex=.8,bty="n")
```

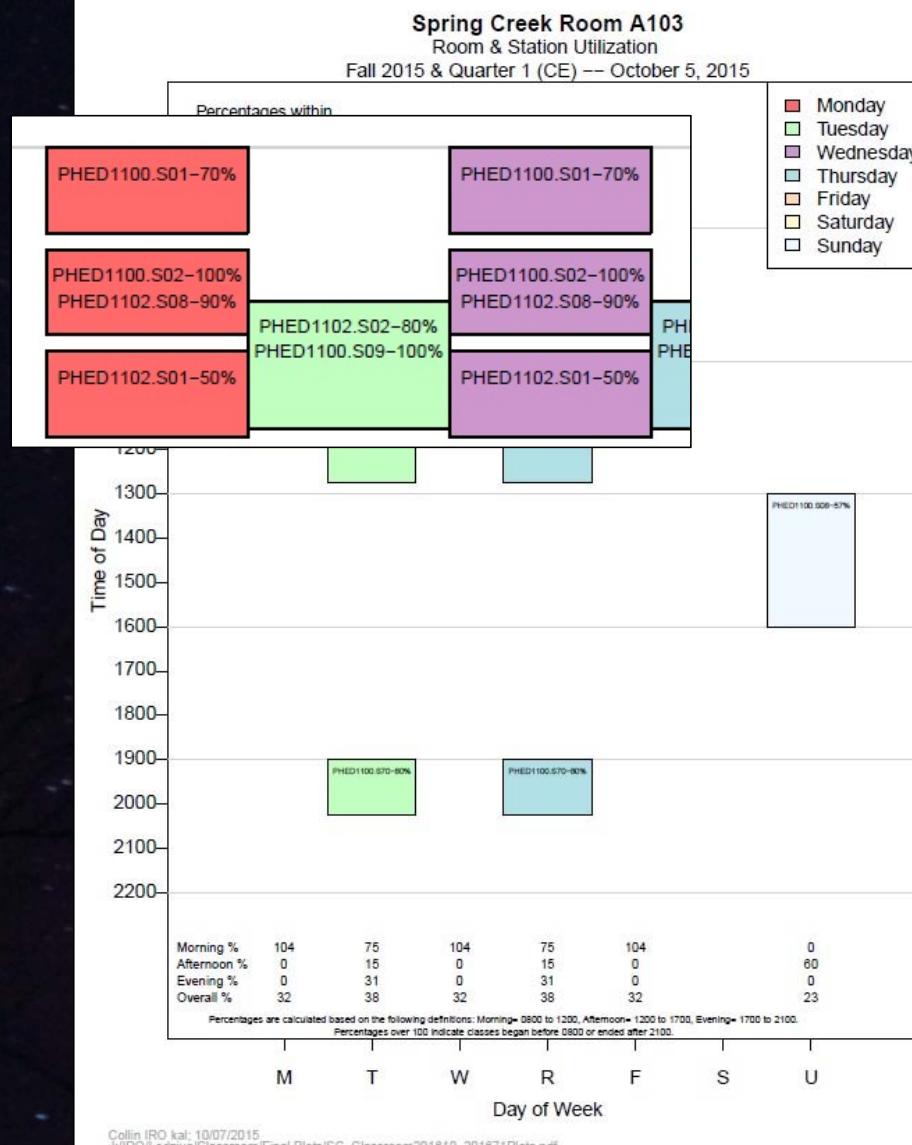


- Using legend function giving coordinates for precise placement.

```
legend(-.3,.68,c("Morning %","Afternoon %","Evening %","Overall %"),bty="n",cex=.51)
```

- Use text function to print definitions.

```
text(3.5,-.2,"Percentages are calculated based on the following  
definitions: Morning= 0800 to 1200, Afternoon= 1200 to 1700,  
Evening= 1700 to 2100.\nPercentages over 100 indicate classes  
began before 0800 or ended after 2100.",cex=.5)
```



- Also use text function to calculate and print station usage.

```
text(Point1+((Point2-Point1)/2), (Point3-(.1*Mucode2)),  
paste(Subj,Crse,".",SEC2,"-",  
round((Enr/Max_Enr)*100),"%",sep=""), cex=.457)
```

Part Two: Heat Map of Campus/Building Usage

Room Usage
Collin College
Fall 2015 (201610)

- Used R to aggregate the data and Excel to create the heat map.

	Spring Creek Campuswide 148 Rooms						
	M	T	W	R	F	S	U
7 am	2%	3%	2%	3%	2%	0%	0%
8 am	30%	50%	32%	51%	27%	7%	0%
9 am	65%	61%	70%	62%	51%	21%	0%
10 am	72%	82%	76%	84%	53%	22%	0%
11 am	72%	89%	73%	90%	50%	24%	0%
12 pm	54%	73%	51%	72%	37%	22%	2%
1 pm	84%	81%	80%	78%	11%	20%	12%
2 pm	84%	86%	83%	85%	9%	19%	12%
3 pm	57%	62%	55%	61%	8%	18%	12%
4 pm	24%	26%	23%	26%	4%	15%	11%
5 pm	38%	39%	34%	38%	3%	5%	5%
6 pm	38%	42%	36%	35%	7%	5%	5%
7 pm	48%	54%	44%	46%	7%	0%	0%
8 pm	47%	51%	44%	44%	7%	0%	0%
9 pm	30%	34%	29%	30%	7%	0%	0%
10 pm	5%	6%	6%	5%	0%	0%	0%
11 pm	3%	4%	3%	4%	0%	0%	0%

Term	Campus	Day	Evening	Day	Ct	T_0700	T_0800	T_0900	T_1000	T_1100	T_1200	T_1300	T_1400	T_1500	T_1600	T_1700	T_1800	T_1900	T_2000	T_2100	T_2200	T_2300	CRN	Time_Begin		
201610	SC	Day	F	1				900																10238	900	
201610	SC	Day	F	1					1000															10294	1000	
201610	SC	Day	M	1					900															10238	900	
201610	SC	Day	M	1					1000															10294	1000	
201610	SC	Day	M	1						1200	1300													50094	1200	
201610	SC	Day	M	1						1300	1400													10306	1300	
201610	SC	Day	M	1						1400	1500													10308	1430	
201610	SC	Evening	M	1								1700	1800											10347	1730	
201610	SC	Evening	M	1								1700	1800											10348	1730	
201610	SC	Day	R	1		800	900																	10295	830	
201610	SC	Day	R	1				1000	1100															10251	1000	
201610	SC	Day	R	1					1100	1200														10248	1130	
201610	SC	Day	R	1					1100	1200														10292	1130	
201610	SC	Day	R	1						1300	1400													10253	1300	
201610	SC	Day	R	1						1300	1400													10299	1300	
201610	SC	Day	R	1								1600	1700											19283	1600	
201610	SC	Day	R	1								1600	1700											19284	1600	
201610	SC	Evening	R	1														1900	2000	2100				10346	1900	
201610	SC	Day	T	1		800	900																	10295	830	
201610	SC	Day	T	1				1000	1100															10251	1000	
201610	SC	Day	T	1					1100	1200														10248	1130	
201610	SC	Day	T	1					1100	1200														10292	1130	
201610	SC	Day	T	1						1300	1400													10253	1300	
201610	SC	Day	T	1						1300	1400													10299	1300	
201610	SC	Day	T	1								1600	1700											19283	1600	
201610	SC	Day	T	1								1600	1700											19284	1600	
201610	SC	Evening	T	1														1900	2000	2100				11226	1900	
201610	SC	Day	W	1			900																		10238	900
201610	SC	Day	W	1				1000																	10294	1000
201610	SC	Day	W	1						1200	1300													50094	1200	
201610	SC	Day	W	1						1300	1400													10306	1300	
201610	SC	Day	W	1							1400	1500												10308	1430	
201610	SC	Evening	W	1									1700	1800											10347	1730
201610	SC	Evening	W	1									1700	1800											10348	1730
201610	SC	Day	F	1	700																				10196	700

- Create a variable to concatenate room, day, and class time so that we can unduplicate for the occurrence of multiple classes meeting in same room at same time.

```
undup7 <- paste(Room_Code,Day_Code,T_0700)
```

- Subset hours so that each room is counted only once per day and time even if there is more than one class meeting at the same time.

```
Undup0700 <- as.numeric(subset(T_0700,!duplicated(undup7)))
```

- Create a variable (Wing) that identifies the wing or building, then concatenate wing and day.

```
t <- paste(Wing ,Day_Code)
```

- Create a vector of all wing and day combinations.

```
tog7 <- subset(t,!duplicated(undup7))
```

- Use the grouping variable to sum the hourly unduplicated hours by wing and day.

```
P7 <- tapply(Undup0700, tog7, sum)
```



Term	Campus	DayEvening	Day	Ct	T_0700	T_0800	T_0900	T_1000	T_1100	T_1200	T_1300	T_1400	T_1500	T_1600	T_1700	T_1800	T_1900	T_2000	T_2100	T_2200	T_2300	CRN	Time_Begin		
201610	SC	Day	F	1			900																10238	900	
201610	SC	Day	F	1				1000															10294	1000	
201610	SC	Day	M	1			900																10238	900	
201610	SC	Day	M	1				1000															10294	1000	
201610	SC	Day	M	1					1200	1300													50094	1200	
201610	SC	Day	M	1					1300	1400													10306	1300	
201610	SC	Day	M	1					1400	1500													10308	1430	
201610	SC	Evening	M	1							1700	1800											10347	1730	
201610	SC	Evening	M	1							1700	1800											10348	1730	
	/	R	1		800	900																	10295	830	
	/	R	1			1000	1100																10251	1000	
	/	R	1				1100	1200															10248	1130	
	/	R	1				1100	1200															10292	1130	
	/	R	1					1300	1400														10253	1300	
	/	R	1					1300	1400														10299	1300	
	/	R	1						1600	1700													19283	1600	
	/	R	1						1600	1700													19284	1600	
																						10346	1900		
																						10295	830		
																						10251	1000		
																						10248	1130		
																						10292	1130		
																						10253	1300		
																						10299	1300		
																						19283	1600		
																						19284	1600		
																						10346	1900		
																						10295	830		
																						10251	1000		
																						10248	1130		
																						10292	1130		
																						10253	1300		
																						10299	1300		
																						19283	1600		
																						19284	1600		
																						11226	1900		
																						10238	900		
																						10294	1000		
																						50094	1200		
																						10306	1300		
																						10308	1430		
																						10347	1730		
																						10348	1730		
																						10196	700		
201610	SC	Day	F	1	700																				
201610	SC	Day	W	1																					
201610	SC	Evening	W	1																					
201610	SC	Evening	W	1																					
201610	SC	Day	F	1	700																				

To indicate activity during each hour, the data has the time placed in the cell, so R code needs to account for that in calculations.

- Create a data frame of daily number of rooms in each wing, building, or campus.

P7 <- as.data.frame(P7/700)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Day	700	800	900	1000	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000	2100	2200	2300	
2	1	3	44	96	106	107	80	125	125	84	36	56	56	71	69	44	8	4	
3	2	5	74	90	122	131	108	120	127	92	39	57	62	80	76	50	9	6	
4	3	3	48	103	113	108	76	119	123	82	34	51	53	65	65	43	9	5	
5	4	5	76	92	124	133	107	116	126	91	39	56	52	68	65	44	8	6	
6	5	3	40	75	79	74	55	17	13	12	6	5	10	11	11	10	0	0	
7	6	0	11	31	33	35	32	30	28	26	22	8	7	0	0	0	0	0	
8	7	0	0	0	0	0	3	18	18	18	16	8	7	0	0	0	0	0	
9																			
10																			
11																			

- Create Pivot Table and use drop-down menu to select buildings or entire campus. Paste counts for each building on individual pages of spreadsheet.

Screenshot of a Microsoft Excel PivotTable setup. The PivotTable Fields pane on the right shows various building names checked under the "VALUES" category. The PivotTable itself displays data for buildings 700 through 2300 across seven categories (1-7) and includes a "Grand Total" column.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2														
3		Column Labels	C	D	E	F	G	H	I	J	K	L	M	N
4	Values	1	2	3	4	5	6	7	Grand Total					
5	Sum of 700	3	5	3	5	3	0	0	19					
6	Sum of 800	44	74	48	76	40	11	0	293					
7	Sum of 900	96	90	103	92	75	31	0	487					
8	Sum of 1000	106	122	113	124	79	33	0	577					
9	Sum of 1100	107	131	108	133	74	35	0	588					
10	Sum of 1200	80	108	76	107	55	32	3	461					
11	Sum of 1300	125	120	119	116	17	30	18	545					
12	Sum of 1400	125	127	123	126	13	28	18	560					
13	Sum of 1500	84	92	82	91	12	26	18	405					
14	Sum of 1600	36	39	34	39	6	22	16	192					
15	Sum of 1700	56	57	51	56	5	8	8	241					
16	Sum of 1800	56	62	53	52	10	7	7	247					
17	Sum of 1900	71	80	65	68	11	0	0	295					
18	Sum of 2000	69	76	65	65	11	0	0	286					
19	Sum of 2100	44	50	43	44	10	0	0	191					
20	Sum of 2200	8	9	9	8	0	0	0	34					
21	Sum of 2300	4	6	5	6	0	0	0	21					
22														
23														
24														
25														
26														
27														
28														
29														
30														
31														

PivotTable Fields
Choose fields to add to report:

- Day
- 700
- 800
- 900
- 1000
- 1100
- 1200
- 1300
- 1400
- 1500
- 1600
- 1700
- 1800
- 1900
- 2000

Drag fields between areas below:

FILTERS

COLUMNS

Day

ROWS

VALUES

Σ Values

Sum of 700 ▾
Sum of 800 ▾
Sum of 900 ▾



Use the *Conditional Formatting* tool to colorize.

Convert to percentages.

The screenshot shows an Excel spreadsheet titled "201610SpringWeekHourlyCampusWide.xlsx". The spreadsheet contains a table of room usage data for "Spring Creek Campuswide" over 148 rooms. The columns represent time intervals from 7 am to 11 pm, and the rows represent different rooms. The data is color-coded using conditional formatting, where darker shades of green and yellow indicate higher usage percentages (e.g., 84% at 1 pm) and lighter shades indicate lower usage percentages (e.g., 0% at 11 pm). The Conditional Formatting dropdown menu is open, highlighting the "Color Scales" option, which is used to apply the color scheme to the data.

	M	T	W	R	F	S	U
7 am	2%	3%	2%	3%	2%	0%	0%
8 am	30%	50%	32%	51%	27%	7%	0%
9 am	65%	61%	70%	62%	51%	21%	0%
10 am	72%	82%	76%	84%	53%	22%	0%
11 am	72%	89%	73%	90%	50%	24%	0%
12 pm	54%	73%	51%	72%	37%	22%	2%
1 pm	84%	81%	80%	78%	11%	20%	12%
2 pm	84%	86%	83%	85%	9%	19%	12%
3 pm	57%	62%	55%	61%	8%	18%	12%
4 pm	24%	26%	23%	26%	4%	15%	11%
5 pm	38%	39%	34%	38%	3%	5%	5%
6 pm	38%	42%	36%	35%	7%	5%	5%
7 pm	48%	54%	44%	46%	7%	0%	0%
8 pm	47%	51%	44%	44%	7%	0%	0%
9 pm	30%	34%	29%	30%	7%	0%	0%
10 pm	5%	6%	6%	5%	0%	0%	0%
11 pm	3%	4%	3%	4%	0%	0%	0%



Kathy Ledzius
kledzius@collin.edu
972-377-1058

Dr. Thomas K. Martin
tmartin@collin.edu
972-758-3817

