

# Untitled

*Chris Leonard*

*September 11, 2016*

```
knitr::opts_chunk$set(echo = TRUE)
```

```
salary <- read.csv("C:\\Users\\Chris\\Desktop\\Temp_grad\\Mungs\\Salaries.csv")
```

```
str(salary)
```

```
## 'data.frame': 397 obs. of 7 variables:
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...
## $ rank : Factor w/ 3 levels "AssocProf","AsstProf",...: 3 3 2 3 3 1 3 3 3 3 ...
## $ discipline : Factor w/ 2 levels "A","B": 2 2 2 2 2 2 2 2 2 2 ...
## $ yrs.since.phd: int 19 20 4 45 40 6 30 45 21 18 ...
## $ yrs.service : int 18 16 3 39 41 6 23 45 20 18 ...
## $ sex : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 1 ...
## $ salary : int 139750 173200 79750 115000 141500 97000 175000 147765 119250 129000 ...
```

```
Salary_Data_Red <- salary[,c(2,6,7)]
```

```
Salary_Data_Red$salary <- as.numeric(Salary_Data_Red$salary)
```

```
Salary_Data_Red$sex <- as.numeric(as.integer(Salary_Data_Red$sex))
```

```
##prof < assoc < assis
```

```
Salary_Data_Red$rank <- as.numeric(as.integer(Salary_Data_Red$rank))
```

```
print(Salary_Data_Red)
```

```
##      rank sex salary
## 1      3   2 139750
## 2      3   2 173200
## 3      2   2  79750
## 4      3   2 115000
## 5      3   2 141500
## 6      1   2  97000
## 7      3   2 175000
## 8      3   2 147765
## 9      3   2 119250
## 10     3   1 129000
## 11     1   2 119800
## 12     2   2  79800
## 13     2   2  77700
## 14     2   2  78000
## 15     3   2 104800
## 16     3   2 117150
## 17     3   2 101000
## 18     3   2 103450
## 19     3   2 124750
## 20     3   1 137000
## 21     3   2  89565
```

## 22	3	2	102580
## 23	3	2	93904
## 24	3	2	113068
## 25	1	1	74830
## 26	3	2	106294
## 27	3	2	134885
## 28	2	2	82379
## 29	2	2	77000
## 30	3	2	118223
## 31	3	2	132261
## 32	2	2	79916
## 33	3	2	117256
## 34	2	2	80225
## 35	2	1	80225
## 36	2	1	77000
## 37	3	2	155750
## 38	2	2	86373
## 39	3	2	125196
## 40	1	2	100938
## 41	3	2	146500
## 42	1	2	93418
## 43	3	2	101299
## 44	3	2	231545
## 45	3	2	94384
## 46	3	2	114778
## 47	3	2	98193
## 48	3	1	151768
## 49	3	1	140096
## 50	2	2	70768
## 51	3	2	126621
## 52	3	2	108875
## 53	2	1	74692
## 54	3	2	106639
## 55	1	2	103760
## 56	1	2	83900
## 57	3	2	117704
## 58	1	2	90215
## 59	1	2	100135
## 60	2	2	75044
## 61	1	2	90304
## 62	2	2	75243
## 63	3	2	109785
## 64	1	1	103613
## 65	2	2	68404
## 66	1	2	100522
## 67	3	2	101000
## 68	3	2	99418
## 69	3	1	111512
## 70	3	2	91412
## 71	3	2	126320
## 72	3	2	146856
## 73	3	2	100131
## 74	3	2	92391
## 75	3	2	113398

## 76	2	2	73266
## 77	3	2	150480
## 78	3	2	193000
## 79	2	2	86100
## 80	2	2	84240
## 81	3	2	150743
## 82	3	2	135585
## 83	3	2	144640
## 84	2	2	88825
## 85	3	1	122960
## 86	3	2	132825
## 87	3	2	152708
## 88	2	2	88400
## 89	3	2	172272
## 90	1	2	107008
## 91	2	1	97032
## 92	1	2	105128
## 93	1	2	105631
## 94	3	2	166024
## 95	3	2	123683
## 96	2	2	84000
## 97	1	2	95611
## 98	3	2	129676
## 99	3	2	102235
## 100	3	2	106689
## 101	3	2	133217
## 102	3	2	126933
## 103	3	2	153303
## 104	3	1	127512
## 105	1	2	83850
## 106	3	2	113543
## 107	1	2	82099
## 108	1	2	82600
## 109	1	2	81500
## 110	3	2	131205
## 111	3	2	112429
## 112	1	2	82100
## 113	2	2	72500
## 114	3	2	104279
## 115	3	1	105000
## 116	3	2	120806
## 117	3	2	148500
## 118	3	2	117515
## 119	2	2	72500
## 120	2	1	73500
## 121	3	2	115313
## 122	3	2	124309
## 123	3	2	97262
## 124	1	1	62884
## 125	3	2	96614
## 126	3	2	78162
## 127	3	2	155500
## 128	2	1	72500
## 129	3	2	113278

## 130	2	2	73000
## 131	1	2	83001
## 132	3	2	76840
## 133	1	1	77500
## 134	2	1	72500
## 135	3	2	168635
## 136	3	2	136000
## 137	3	2	108262
## 138	3	2	105668
## 139	1	2	73877
## 140	3	2	152664
## 141	1	2	100102
## 142	1	2	81500
## 143	3	2	106608
## 144	2	2	89942
## 145	3	2	112696
## 146	3	2	119015
## 147	2	2	92000
## 148	3	2	156938
## 149	3	1	144651
## 150	2	2	95079
## 151	3	2	128148
## 152	2	2	92000
## 153	3	2	111168
## 154	1	1	103994
## 155	2	2	92000
## 156	3	2	118971
## 157	1	2	113341
## 158	2	2	88000
## 159	1	2	95408
## 160	3	2	137167
## 161	2	2	89516
## 162	3	2	176500
## 163	1	2	98510
## 164	2	2	89942
## 165	2	2	88795
## 166	3	2	105890
## 167	3	2	167284
## 168	3	2	130664
## 169	1	2	101210
## 170	3	2	181257
## 171	2	2	91227
## 172	3	2	151575
## 173	3	2	93164
## 174	3	2	134185
## 175	1	2	105000
## 176	3	2	111751
## 177	1	2	95436
## 178	1	2	100944
## 179	3	2	147349
## 180	2	1	92000
## 181	3	2	142467
## 182	3	2	141136
## 183	1	2	100000

## 184	3	2	150000
## 185	3	2	101000
## 186	3	2	134000
## 187	1	1	103750
## 188	3	2	107500
## 189	1	2	106300
## 190	3	2	153750
## 191	3	2	180000
## 192	3	2	133700
## 193	3	2	122100
## 194	1	2	86250
## 195	1	2	90000
## 196	1	2	113600
## 197	2	2	92700
## 198	2	2	92000
## 199	3	2	189409
## 200	3	2	114500
## 201	2	2	92700
## 202	3	2	119700
## 203	3	2	160400
## 204	3	2	152500
## 205	3	2	165000
## 206	3	2	96545
## 207	3	2	162200
## 208	3	2	120000
## 209	2	2	91300
## 210	3	2	163200
## 211	2	2	91000
## 212	3	2	111350
## 213	3	2	128400
## 214	3	2	126200
## 215	1	2	118700
## 216	3	2	145350
## 217	3	2	146000
## 218	1	2	105350
## 219	1	1	109650
## 220	3	2	119500
## 221	3	2	170000
## 222	3	2	145200
## 223	1	2	107150
## 224	3	2	129600
## 225	3	2	87800
## 226	3	2	122400
## 227	2	2	63900
## 228	1	2	70000
## 229	3	2	88175
## 230	3	2	133900
## 231	3	1	91000
## 232	1	1	73300
## 233	3	2	148750
## 234	3	1	117555
## 235	2	2	69700
## 236	3	2	81700
## 237	3	2	114000

##	238	2	1	63100
##	239	3	2	77202
##	240	3	2	96200
##	241	2	2	69200
##	242	3	2	122875
##	243	3	2	102600
##	244	3	2	108200
##	245	3	2	84273
##	246	3	1	90450
##	247	3	2	91100
##	248	3	2	101100
##	249	3	2	128800
##	250	3	2	204000
##	251	3	2	109000
##	252	3	2	102000
##	253	3	2	132000
##	254	2	1	77500
##	255	3	1	116450
##	256	1	2	83000
##	257	3	2	140300
##	258	1	2	74000
##	259	2	2	73800
##	260	3	2	92550
##	261	1	2	88600
##	262	3	2	107550
##	263	3	2	121200
##	264	3	2	126000
##	265	3	2	99000
##	266	3	2	134800
##	267	3	2	143940
##	268	3	2	104350
##	269	3	2	89650
##	270	3	2	103700
##	271	3	2	143250
##	272	3	2	194800
##	273	2	2	73000
##	274	2	2	74000
##	275	2	1	78500
##	276	3	2	93000
##	277	3	2	107200
##	278	3	2	163200
##	279	3	2	107100
##	280	3	2	100600
##	281	3	2	136500
##	282	3	2	103600
##	283	3	2	57800
##	284	3	2	155865
##	285	1	2	88650
##	286	1	2	81800
##	287	3	2	115800
##	288	2	2	85000
##	289	3	2	150500
##	290	2	2	74000
##	291	3	2	174500

##	292	3	2	168500
##	293	3	2	183800
##	294	1	2	104800
##	295	3	2	107300
##	296	3	2	97150
##	297	3	2	126300
##	298	3	2	148800
##	299	3	2	72300
##	300	1	2	70700
##	301	3	2	88600
##	302	3	2	127100
##	303	3	2	170500
##	304	3	2	105260
##	305	3	2	144050
##	306	3	2	111350
##	307	2	2	74500
##	308	3	2	122500
##	309	2	2	74000
##	310	3	2	166800
##	311	3	2	92050
##	312	3	2	108100
##	313	3	2	94350
##	314	3	2	100351
##	315	3	2	146800
##	316	2	2	84716
##	317	1	1	71065
##	318	3	2	67559
##	319	3	2	134550
##	320	3	2	135027
##	321	3	2	104428
##	322	1	2	95642
##	323	1	2	126431
##	324	3	1	161101
##	325	3	2	162221
##	326	2	2	84500
##	327	3	2	124714
##	328	3	2	151650
##	329	1	2	99247
##	330	3	2	134778
##	331	3	2	192253
##	332	3	2	116518
##	333	3	1	105450
##	334	3	2	145098
##	335	1	1	104542
##	336	3	2	151445
##	337	3	2	98053
##	338	3	2	145000
##	339	3	2	128464
##	340	3	2	137317
##	341	3	2	106231
##	342	3	1	124312
##	343	3	2	114596
##	344	3	2	162150
##	345	3	2	150376

##	346	3	2	107986
##	347	3	2	142023
##	348	3	2	128250
##	349	2	2	80139
##	350	3	2	144309
##	351	3	2	186960
##	352	3	2	93519
##	353	3	2	142500
##	354	3	2	138000
##	355	2	2	83600
##	356	3	2	145028
##	357	3	2	88709
##	358	3	2	107309
##	359	3	1	109954
##	360	2	2	78785
##	361	3	2	121946
##	362	3	1	109646
##	363	3	2	138771
##	364	1	2	81285
##	365	3	2	205500
##	366	3	2	101036
##	367	3	2	115435
##	368	1	2	108413
##	369	3	2	131950
##	370	3	2	134690
##	371	1	2	78182
##	372	3	2	110515
##	373	3	2	109707
##	374	3	2	136660
##	375	3	2	103275
##	376	3	2	103649
##	377	2	2	74856
##	378	2	2	77081
##	379	3	2	150680
##	380	1	2	104121
##	381	2	2	75996
##	382	3	2	172505
##	383	1	2	86895
##	384	3	2	105000
##	385	3	2	125192
##	386	3	2	114330
##	387	3	2	139219
##	388	3	2	109305
##	389	3	2	119450
##	390	3	2	186023
##	391	3	2	166605
##	392	3	2	151292
##	393	3	2	103106
##	394	3	2	150564
##	395	3	2	101738
##	396	3	2	95329
##	397	2	2	81035



```
analysis1 <-aggregate(Salary_Data_Red, by=list(Salary_Data_Red$rank, Salary_Data_Red$sex), mean)
paste(analysis1)
```

```
## [1] "c(1, 2, 3, 1, 2, 3)"
## [2] "c(1, 1, 1, 2, 2, 2)"
## [3] "c(1, 2, 3, 1, 2, 3)"
## [4] "c(1, 1, 1, 2, 2, 2)"
## [5] "c(88512.8, 78049.9090909091, 121967.6111111111, 94869.7037037037, 81311.4642857143, 127120.8225806452)"
```

```
str(Salary_Data_Red)
```

```
## 'data.frame': 397 obs. of 3 variables:
## $ rank : num 3 3 2 3 3 1 3 3 3 3 ...
## $ sex : num 2 2 2 2 2 2 2 2 2 1 ...
## $ salary: num 139750 173200 79750 115000 141500 ...
```

```
aggregate(Salary_Data_Red, by=list(Salary_Data_Red$rank, Salary_Data_Red$sex), mean)
```

```
## Group.1 Group.2 rank sex salary
## 1 1 1 1 1 88512.80
## 2 2 1 2 1 78049.91
## 3 3 1 3 1 121967.61
## 4 1 2 1 2 94869.70
## 5 2 2 2 2 81311.46
## 6 3 2 3 2 127120.82
```

```
MaxSal <- aggregate(Salary_Data_Red, by=list(Salary_Data_Red$rank, Salary_Data_Red$sex), max)
MinSal <- aggregate(Salary_Data_Red, by=list(Salary_Data_Red$rank, Salary_Data_Red$sex), min)
##function to measure salary range
gap <- function(x) {
  g <- max(x) -min(x)
  return(g)
}
SalGap <- aggregate(Salary_Data_Red, by=list(Salary_Data_Red$rank, Salary_Data_Red$sex), gap)
paste(SalGap)
```

```
## [1] "c(1, 2, 3, 1, 2, 3)"
## [2] "c(1, 1, 1, 2, 2, 2)"
## [3] "c(0, 0, 0, 0, 0, 0)"
## [4] "c(0, 0, 0, 0, 0, 0)"
## [5] "c(46766, 33932, 70651, 56431, 31179, 173745)"
```

```
stacklist <-aggregate(salary ~ rank + sex, salary, summary)
```