



UC San Diego  
HEALTH SCIENCES  
CLINICAL AND TRANSLATIONAL  
RESEARCH INSTITUTE

---

Statistical Analysis Report  
CTRI Protocol 8286

# Report of the UC San Diego Health Sciences Task Force on Gender Equity

---

January 16, 2015  
Investigator: Steven Plaxe  
Supervising Statistician: Ronghui Xu  
Statistician: Jiayi Hou  
Statistician: Lishi Zhang

### 4.3 R Output and Diagnostics

Model 1:  $\log(\text{total salary}_{ij}) = \text{Department}_{ij} + \text{Series}_{ij} + \text{Degree}_{ij} + \text{Rank}_{ij} + \text{Gender}_{ij} + \text{Years of experience at UCSD}_{ij} + \text{Years of experience at UCSD}_{ij}^2 + \text{Fiscal year}_{ij} + \mathbf{u}_i + \epsilon_{ij}$

```
Linear mixed model fit by maximum likelihood ['lmerMod']
Formula: log(adj.salary) ~ Rank + Gender + Series + HomeDept + c_first_appt +
  year + Degree + I(c_first_appt^2) + (1 | EMPLOYEE_ID)
Data: na.omit(newdat2)
```

AIC	BIC	logLik	deviance	df.resid
-5044.3	-4774.7	2562.1	-5124.3	6199

Scaled residuals:

Min	1Q	Median	3Q	Max
-8.1545	-0.4028	-0.0249	0.3916	7.1438

Random effects:

Groups	Name	Variance	Std.Dev.
EMPLOYEE_ID	(Intercept)	0.09564	0.3093
	Residual	0.01049	0.1024

Number of obs: 6239, groups: EMPLOYEE\_ID, 1607

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	-2.482e+01	2.637e+00	-9.411
RankAssoc Prof	6.298e-02	7.436e-03	8.469
RankProfessor	1.281e-01	1.109e-02	11.551
GenderF	-1.231e-01	1.706e-02	-7.213
SeriesADJ	-2.698e-01	1.954e-02	-13.811
SeriesCLIN	-1.617e-01	2.009e-02	-8.050
SeriesCLINx	-1.294e-01	2.212e-02	-5.852
SeriesIR	-1.334e-01	1.830e-02	-7.289
HomeDeptBIOENGINEERING	7.373e-02	3.163e-01	0.233
HomeDeptCELL & MOL MED PROG	-5.013e-02	9.680e-02	-0.518
HomeDeptCHEMISTRY & BIOCHEMISTRY	1.490e-01	1.865e-01	0.799
HomeDeptDEP. OF REPRODUCTIVE MEDICINE	-4.690e-02	6.106e-02	-0.768
HomeDeptDIVISION OF BIOLOGICAL SCI.	5.508e-02	3.158e-01	0.174
HomeDeptFAMILY & PREVENTIVE MEDICINE	-2.444e-01	4.896e-02	-4.992
HomeDeptMEDICINE	-2.662e-01	3.927e-02	-6.779
HomeDeptMEDSCH/EMERG MED SVC	-1.139e-01	6.991e-02	-1.629
HomeDeptNEUROSCIENCES	-2.160e-01	5.121e-02	-4.218
HomeDeptOPHTHALMOLOGY	1.819e-03	7.009e-02	0.026
HomeDeptORTHOPAEDIC SURGERY	2.471e-01	6.551e-02	3.771
HomeDeptPATHOLOGY	-1.465e-01	5.340e-02	-2.744
HomeDeptPEDIATRICS	-1.557e-01	4.321e-02	-3.603
HomeDeptPHARMACOLOGY	-1.633e-01	8.010e-02	-2.039
HomeDeptPSYCHIATRY	-2.559e-01	4.643e-02	-5.512
HomeDeptRADIATION MED & APPLIED SCI	2.195e-01	7.576e-02	2.898
HomeDeptRADIOLOGY	-4.484e-02	5.182e-02	-0.865
HomeDeptSCH OF PHARMACY AND PHARM. SCI	-1.316e-01	7.781e-02	-1.691
HomeDeptSURGERY	1.693e-01	4.646e-02	3.645
c_first_appt	6.437e-03	9.557e-04	6.735
year	1.859e-02	1.309e-03	14.205
DegreeDO/PhD	-2.183e-01	3.268e-01	-0.668
DegreeDPM	-4.153e-01	2.357e-01	-1.762
DegreeMD	1.197e-01	6.980e-02	1.715
DegreeMD/PhD	-3.894e-02	7.482e-02	-0.520
DegreeNonDoc	-4.138e-01	3.338e-01	-1.240
DegreeOtherDoc	-1.191e-01	2.367e-01	-0.503
DegreePharmD	-3.580e-01	1.160e-01	-3.088
DegreePhD	-4.094e-01	7.134e-02	-5.739
I(c_first_appt^2)	-2.139e-04	4.517e-05	-4.736

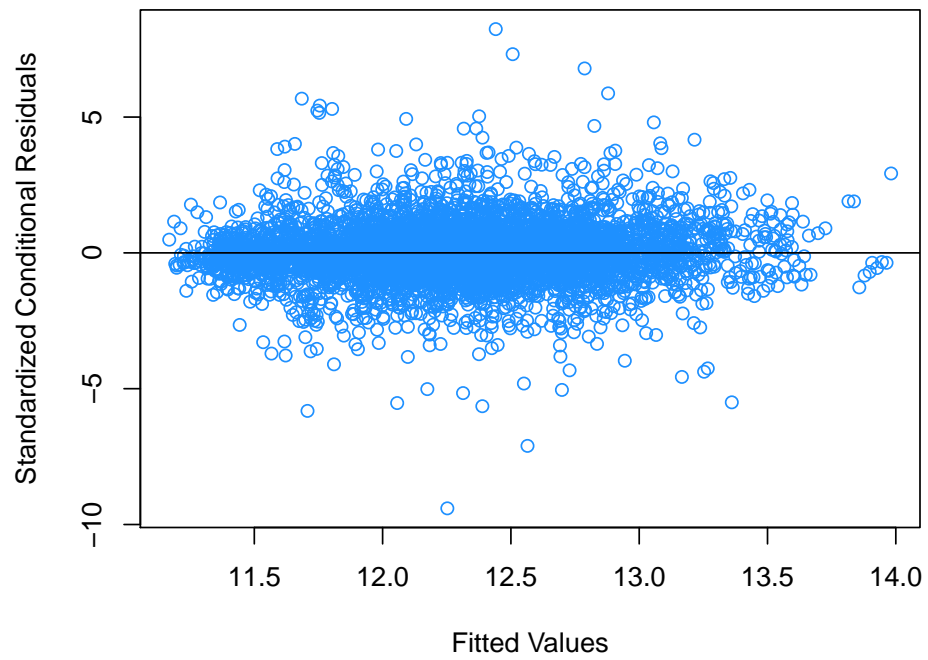


Figure 10: Standardized conditional residuals for the mixed model fit for total salary with URM as the predictor of interest.

Model 2:  $\log(\text{total salary}_{ij}) = \text{Department}_{ij} + \text{Series}_{ij} + \text{Degree}_{ij} + \text{Rank}_{ij} + \text{URM}_{ij} + \text{Years of experience at UCSD}_{ij} + \text{Years of experience at UCSD}_{ij}^2 + \text{Fiscal year}_{ij} + \mathbf{u}_i + \epsilon_{ij}$

```
Linear mixed model fit by maximum likelihood ['lmerMod']
Formula: log(adj.salary) ~ Rank + URM + Series + HomeDept + c_first_appt +
  year + Degree + I(c_first_appt^2) + (1 | EMPLOYEE_ID)
Data: na.omit(newdat2)
```

AIC	BIC	logLik	deviance	df.resid
-4995.2	-4725.6	2537.6	-5075.2	6199

Scaled residuals:

Min	1Q	Median	3Q	Max
-8.1590	-0.4036	-0.0263	0.3969	7.1562

Random effects:

Groups	Name	Variance	Std.Dev.
EMPLOYEE_ID	(Intercept)	0.09816	0.3133
Residual		0.01051	0.1025

Number of obs: 6239, groups: EMPLOYEE\_ID, 1607

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	-2.356e+01	2.647e+00	-8.904
RankAssoc Prof	6.498e-02	7.445e-03	8.728
RankProfessor	1.324e-01	1.109e-02	11.932
URMURM	-4.771e-02	3.061e-02	-1.559
SeriesADJ	-2.789e-01	1.958e-02	-14.240
SeriesCLIN	-1.784e-01	2.009e-02	-8.879
SeriesCLINx	-1.411e-01	2.219e-02	-6.357
SeriesIR	-1.374e-01	1.837e-02	-7.482
HomeDeptBIOENGINEERING	6.655e-02	3.204e-01	0.208
HomeDeptCELL & MOL MED PROG	-5.068e-02	9.804e-02	-0.517
HomeDeptCHEMISTRY & BIOCHEMISTRY	1.660e-01	1.889e-01	0.879
HomeDeptDEP. OF REPRODUCTIVE MEDICINE	-7.781e-02	6.168e-02	-1.261
HomeDeptDIVISION OF BIOLOGICAL SCI.	6.684e-02	3.199e-01	0.209
HomeDeptFAMILY & PREVENTIVE MEDICINE	-2.757e-01	4.939e-02	-5.582
HomeDeptMEDICINE	-2.827e-01	3.970e-02	-7.121
HomeDeptMEDSCH/EMERG MED SVC	-1.069e-01	7.083e-02	-1.509
HomeDeptNEUROSCIENCES	-2.251e-01	5.185e-02	-4.341
HomeDeptOPHTHALMOLOGY	-2.146e-03	7.099e-02	-0.030
HomeDeptORTHOPAEDIC SURGERY	2.525e-01	6.636e-02	3.804
HomeDeptPATHOLOGY	-1.720e-01	5.397e-02	-3.187
HomeDeptPEDIATRICS	-1.926e-01	4.344e-02	-4.434
HomeDeptPHARMACOLOGY	-1.895e-01	8.102e-02	-2.339
HomeDeptPSYCHIATRY	-2.834e-01	4.687e-02	-6.047
HomeDeptRADIATION MED & APPLIED SCI	2.294e-01	7.671e-02	2.990
HomeDeptRADIOLOGY	-5.861e-02	5.247e-02	-1.117
HomeDeptSCH OF PHARMACY AND PHARM. SCI	-1.574e-01	7.869e-02	-2.001
HomeDeptSURGERY	1.653e-01	4.704e-02	3.515
c_first_appt	6.843e-03	9.650e-04	7.091
year	1.796e-02	1.314e-03	13.672
DegreeDO/PhD	-1.638e-01	3.308e-01	-0.495
DegreeDPM	-3.499e-01	2.391e-01	-1.464
DegreeMD	1.316e-01	7.067e-02	1.861
DegreeMD/PhD	-1.430e-02	7.571e-02	-0.189
DegreeNonDoc	-3.676e-01	3.377e-01	-1.089
DegreeOtherDoc	-1.780e-01	2.395e-01	-0.743
DegreePharmD	-3.479e-01	1.174e-01	-2.963
DegreePhD	-4.067e-01	7.224e-02	-5.630
I(c_first_appt^2)	-2.070e-04	4.535e-05	-4.563

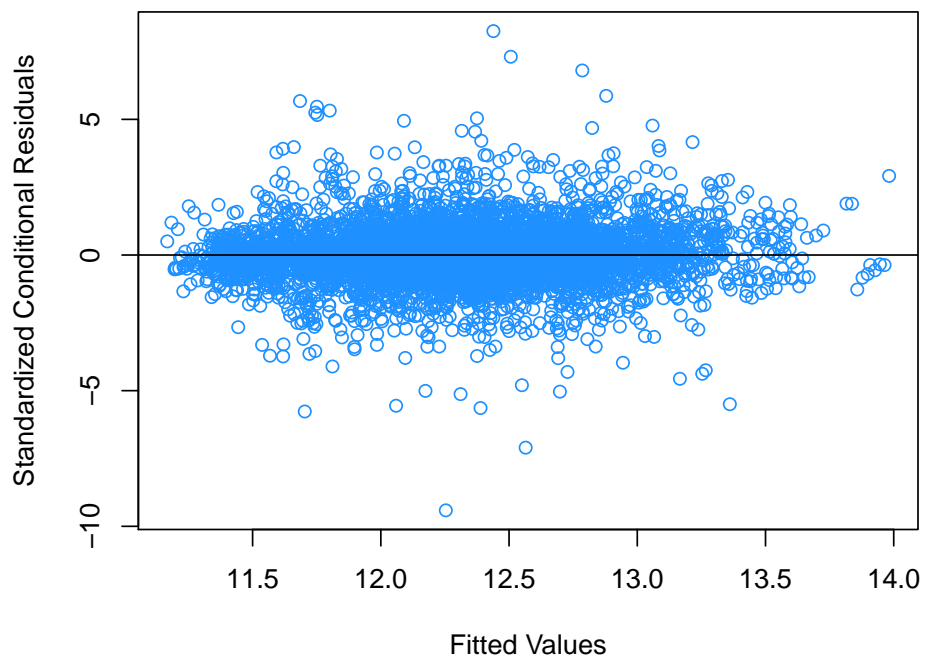


Figure 11: Standardized conditional residuals for the mixed model fit for total salary with gender as the predictor of interest.

Model 3:  $\log(\text{total salary}_{ij}) = \text{Department}_{ij} + \text{Series}_{ij} + \text{Degree}_{ij} + \text{Rank}_{ij} + \text{Gender}_{ij} + \text{URM}_{ij} + \text{Gender}_{ij} * \text{URM}_{ij} + \text{Years of experience at UCSD}_{ij} + \text{Years of experience at UCSD}_{ij}^2 + \text{Fiscal year}_{ij} + \mathbf{u}_i + \epsilon_{ij}$

```
Linear mixed model fit by maximum likelihood ['lmerMod']
Formula: log(adj.salary) ~ Rank + Gender + URM + Gender * URM + Series +
  HomeDept + c_first_appt + year + Degree + I(c_first_appt^2) + (1 | EMPLOYEE_ID)
Data: na.omit(newdat2)
```

AIC	BIC	logLik	deviance	df.resid
-5046.5	-4763.5	2565.2	-5130.5	6197

Scaled residuals:

Min	1Q	Median	3Q	Max
-8.1556	-0.4028	-0.0246	0.3915	7.1435

Random effects:

Groups	Name	Variance	Std.Dev.
EMPLOYEE_ID	(Intercept)	0.09531	0.3087
Residual		0.01049	0.1024

Number of obs: 6239, groups: EMPLOYEE\_ID, 1607

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	-2.488e+01	2.636e+00	-9.440
RankAssoc Prof	6.274e-02	7.435e-03	8.438
RankProfessor	1.278e-01	1.108e-02	11.528
GenderF	-1.324e-01	1.771e-02	-7.477
URMURM	-9.794e-02	4.021e-02	-2.436
SeriesADJ	-2.692e-01	1.953e-02	-13.786
SeriesCLIN	-1.608e-01	2.007e-02	-8.011
SeriesCLINx	-1.281e-01	2.211e-02	-5.796
SeriesIR	-1.329e-01	1.829e-02	-7.264
HomeDeptBIOENGINEERING	6.723e-02	3.158e-01	0.213
HomeDeptCELL & MOL MED PROG	-5.515e-02	9.666e-02	-0.571
HomeDeptCHEMISTRY & BIOCHEMISTRY	1.433e-01	1.862e-01	0.770
HomeDeptDEP. OF REPRODUCTIVE MEDICINE	-4.632e-02	6.096e-02	-0.760
HomeDeptDIVISION OF BIOLOGICAL SCI.	4.956e-02	3.153e-01	0.157
HomeDeptFAMILY & PREVENTIVE MEDICINE	-2.421e-01	4.893e-02	-4.949
HomeDeptMEDICINE	-2.674e-01	3.921e-02	-6.821
HomeDeptMEDSCH/EMERG MED SVC	-1.110e-01	6.983e-02	-1.589
HomeDeptNEUROSCIENCES	-2.179e-01	5.113e-02	-4.262
HomeDeptOPHTHALMOLOGY	-6.439e-04	6.999e-02	-0.009
HomeDeptORTHOAEDIC SURGERY	2.418e-01	6.544e-02	3.695
HomeDeptPATHOLOGY	-1.479e-01	5.332e-02	-2.773
HomeDeptPEDIATRICS	-1.556e-01	4.314e-02	-3.607
HomeDeptPHARMACOLOGY	-1.621e-01	7.997e-02	-2.027
HomeDeptPSYCHIATRY	-2.588e-01	4.636e-02	-5.582
HomeDeptRADIATION MED & APPLIED SCI	2.122e-01	7.569e-02	2.803
HomeDeptRADIOLOGY	-5.052e-02	5.178e-02	-0.976
HomeDeptSCH OF PHARMACY AND PHARM. SCI	-1.306e-01	7.768e-02	-1.681
HomeDeptSURGERY	1.673e-01	4.639e-02	3.606
c_first_appt	6.419e-03	9.547e-04	6.723
year	1.863e-02	1.308e-03	14.238
DegreeDO/PhD	-2.207e-01	3.262e-01	-0.677
DegreeDPM	-3.676e-01	2.361e-01	-1.557
DegreeMD	1.245e-01	6.972e-02	1.786
DegreeMD/PhD	-3.506e-02	7.472e-02	-0.469
DegreeNonDoc	-4.161e-01	3.332e-01	-1.249
DegreeOtherDoc	-1.141e-01	2.364e-01	-0.483
DegreePharmD	-3.495e-01	1.158e-01	-3.018
DegreePhD	-4.060e-01	7.125e-02	-5.698
I(c_first_appt^2)	-2.172e-04	4.517e-05	-4.809
GenderF:URMURM	1.220e-01	6.072e-02	2.009

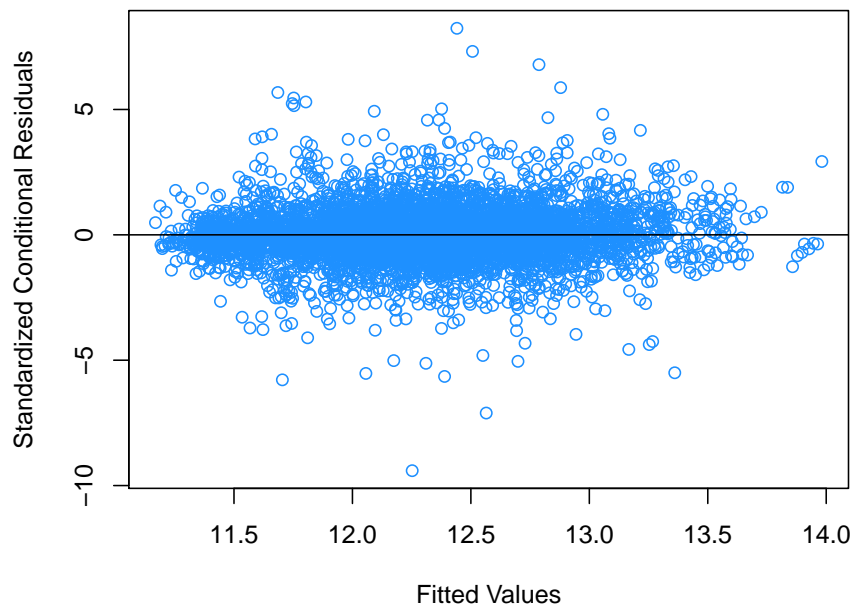


Figure 12: Standardized conditional residuals for the mixed model fit for total salary with gender, URM and the interaction as the predictors of interest.

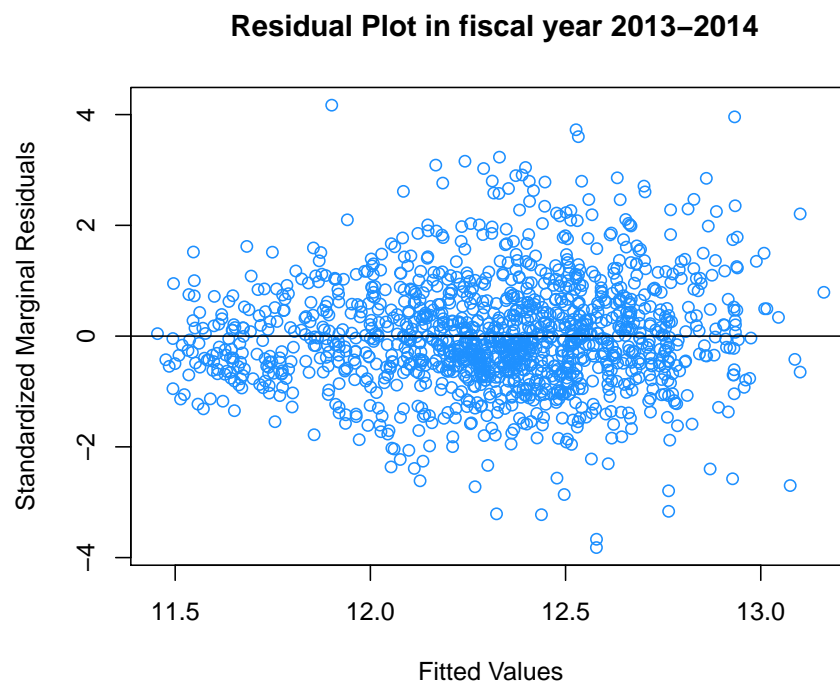


Figure 13: Standardized marginal residuals for the mixed model fit for total salary with gender, URM and the interaction as the predictors of interest.



## 4.4 R Code

```

1 library(coin)
2 library(Hmisc)
3 library(nlme)
4 library(lme4)
5 library(scales)
6 library(ggplot2)
7 library(plyr)
8 library(HLMdiag)
9 library(ggthemes)
10 library(stringr)
11 setwd("~/Users/liz076/Desktop/UCSD/CTRI/Projects/8286_Plaxe")
12 options(digit = 2, stringsAsFactors = FALSE, width = 100, format = 'f')
13
14 ###functions
15 pvalFormat <- function(p.values, method = 'none', replace = FALSE, math = TRUE){
16   ## Formats p-values for reports, can report adjusted pvalues
17   ##   Inputs:
18   ##     - p.value: numeric p-value
19   ##     - method: pvalue adjustment, passed to p.adjust.methods
20   ##     - replace: if TRUE, replaces p-values with their adjusted value
21   ##   Outputs:
22   ##     - out: formatted p-value
23
24   p.values <- as.numeric(p.values)
25   out <- rep(NA, length(p.values))
26   sig <- p.adjust(p.values, method)
27   if(replace) p.values <- sig
28
29   for(i in 1:length(p.values)){
30     if(is.na(p.values[i])){out[i] <- NA}else{
31       if(p.values[i] >= .001){
32         out[i] <- paste('$', formatC(p.values[i], format = 'f', digits = 3), '$', sep = '')
33       }
34
35       if(p.values[i] < .001){
36         out[i] <- '<$.001$'
37       }
38
39       if(sig[i] > 0.01 & sig[i] <= 0.05){
40         out[i] <- paste(out[i], '*', sep = '')
41       }
42
43       if(sig[i] > 0.001 & sig[i] <= 0.01) {
44         out[i] <- paste(out[i], '**', sep = '')
45       }
46
47       if(sig[i] <= 0.001){
48         out[i] <- paste(out[i], '***', sep = '')
49       }
50     }
51   }
52   return(out)
53 }
54
55 demoTab2 <- function(vars, group = NULL, DATA, overall = TRUE, do.test = TRUE, tests = NULL, arg = NULL, FUNs = NULL,
56   ...){
57   ## A function to recreate the tables generated by summary.formula()
58   ##   Inputs:
59   ##     - vars: A character or numeric vector indicating the data columns of interest.
60   ##     - group: A character or numeric vector indicating the groups, if any, to analyze
61
62   makeTab <- function(X, GRPS, do.test, TEST, FUN, ARG, ...){
63     if(class(X) %in% c('integer', 'numeric')){
64       if(!is.null(GRPS)){
65         if(is.null(FUN)){
66           outLine <- c(paste('$', formatC(tapply(X, GRPS, mean, na.rm = T), format = 'f', ...), '\\; ('),
67             formatC(tapply(X, GRPS, sd, na.rm = T), format = 'f', ...), ')$ ', sep = ''),
68             paste('$', formatC(mean(X, na.rm = T), format = 'f', ...), '\\; ('),
69             formatC(sd(X, na.rm = T), format = 'f', ...), ')$ ', sep = ''))
70         if(overall == FALSE) outLine <- outLine[-(length(unique(GRPS)) + 1)]
71       }
72       if(do.test){
73         if(is.null(TEST)){
74           outLine <- c(outLine, pvalFormat(t.test(X ~ GRPS)$p.value))
75         } else{

```

```

76     if(is.null(ARG[[1]])){
77         outLine <- c(outLine, pvalFormat(do.call(TEST, list(as.formula(X ~ GRPS)))$p.value))
78     } else{
79         ARG <- c(X ~ GRPS, ARG); names(ARG)[1] <- 'formula'
80         outLine <- c(outLine, pvalFormat(do.call(TEST, ARG)$p.value))
81     }
82 }
83 }
84 } else{
85     if(is.null(FUN)){
86         outLine <- paste('$', formatC(mean(X), format = 'f', ...), '\\; (' ,
87                                     formatC(sd(X), format = 'f', ...), ')$', sep = '')
88     }
89 }
90 }
91 }
92 if(class(X) %in% c('character', 'factor')){
93     p.test <- NULL
94     if(is.null(GRPS)){
95         if(is.null(FUN)){
96             outLine <- suppressWarnings(paste(table(X), '\\; (' ,
97                                             formatC(prop.table(table(X), margin = 1) * 100, digits = 1, format = 'f') ,
98                                             '\\%', ')', sep = ''))
99         }
100        outLine <- as.matrix(paste('$', outLine, '$', sep = ''), ncol = 1)
101    } else{
102        if(is.null(FUN)){
103            outLine <- paste('$', table(X, GRPS), '\\; (' ,
104                            suppressWarnings(formatC(prop.table(table(X, GRPS), margin = 1) * 100,
105                                                    digits = 1, format = 'f') , '\\%', ')$', sep = ''))
106            outLine2 <- paste('$', table(X, GRPS), '\\; ', sep = '')
107            outLine2 <- cbind(matrix(outLine2, ncol = length(unique(GRPS))),
108                             paste('$', table(X), '\\; ', sep = ''))
109            outLine <- cbind(matrix(outLine, ncol = length(unique(GRPS))),
110                             paste('$', table(X), '\\; (' ,
111                                 suppressWarnings(formatC(prop.table(table(X), margin = 1) * 100, digits = 1, format =
112                                     'f') , '\\%', ')$', sep = ''))
113        }
114        outLine <- outLine[, -ncol(outLine)]
115        outLine = cbind(outLine, outLine2[, ncol(outLine2)])
116    }
117    if(do.test & nrow(outLine) != 1){
118        if(is.null(TEST)){
119            p.test <- c(pvalFormat(chisq.test(table(X, GRPS), correct = FALSE)$p.value), rep('', nrow(outLine)))
120        } else{
121            if(is.null(ARG[[1]])){
122                p.test <- c(pvalFormat(do.call(TEST, list(table(X, GRPS)))$p.value), rep('', nrow(outLine)))
123            } else{
124                ARG$x <- table(X, GRPS)
125                p.test <- c(pvalFormat(do.call(TEST, ARG)$p.value), rep('', nrow(outLine)))
126            }
127        }
128    } else{
129        p.test <- c('-', '')
130    }
131    outLine <- rbind(rep('', ncol(outLine)), outLine)
132    if(!is.null(p.test)) outLine <- cbind(outLine, p.test)
133 }
134 }
135 }
136 if(!is.null(group)){
137     if(class(group) %in% c('factor', 'character')) GROUPS <- which(names(DATA) == group)
138     if(class(group) == 'integer') GROUPS <- group
139 } else{GROUPS <- NULL}
140 }
141 if(!class(vars) %in% c('formula', 'character', 'integer')) stop('Unsupported class for vars argument.')
142 if(class(vars) == 'formula'){
143     GROUPS <- which(names(DATA) == as.character(vars[2]))
144     VAR_NAMES <- strsplit(as.character(vars[3]), '+', fixed = TRUE)[[1]]
145     WHICH <- numeric(length(VAR_NAMES))
146     for(ii in 1:length(WHICH)) WHICH[ii] <- which(names(DATA) == VAR_NAMES[ii])
147 }
148 }
149 if(class(vars) == 'character') WHICH <- which(names(DATA) %in% vars)
150 if(class(vars) == 'integer') WHICH <- vars
151 }
152 if(!is.null(GROUPS)) GRPS <- DATA[, GROUPS] else{GRPS <- NULL}

```

```

153 out <- NULL
154 R NAMES <- NULL
155 if (length(do.test) == 1) do.test <- rep(do.test, length(WHICH))
156 for (i in 1:length(WHICH)){
157   out <- rbind(out, makeTab(DATA[, WHICH[i]], GRPS, do.test[i], tests[i], FUNS[i, ], ARG = arg[[i]], ...))
158   if (class(DATA[, WHICH[i]]) %in% c('integer', 'numeric')) R NAMES <- c(R NAMES, names(DATA)[WHICH[i]])
159   if (class(DATA[, WHICH[i]]) %in% c('character', 'factor')){
160     R NAMES <- c(R NAMES, names(DATA)[WHICH[i]], paste('~::~', names(table(DATA[, WHICH[i]])), sep = ''))
161   }
162 }
163 rownames(out) <- R NAMES
164 if (is.null(GROUPS)) colnames(out) <- paste('Total ($n=', nrow(DATA), '$)', sep = '')
165 if (!is.null(GROUPS)){
166   if (overall == TRUE){
167     colnames(out) <- c(paste(sort(unique(GRPS)), '$n=', table(GRPS), '$)', sep = ''), paste('Total ($n=', nrow(
168       DATA), '$)', sep = ''),
169       'p-value')
170   } else {colnames(out) <- c(paste(sort(unique(GRPS)), '$n=', table(GRPS), '$)', sep = ''), 'p-value')}
171 }
172 return(out)
173 }
174
175 demoTab <- function(vars, group = NULL, DATA, overall = TRUE, do.test = TRUE, tests = NULL, arg = NULL, FUNS = NULL,
176   ...){
177   ## A function to recreate the tables generated by summary.formula()
178   ## Inputs:
179   ## - vars: A character or numeric vector indicating the data columns of interest.
180   ## - Also accepts a formula object, similar to summary.formula.
181   ## - group: A character or numeric vector indicating the groups, if any, to analyze
182   makeTab <- function(X, GRPS, do.test, TEST, FUN, ARG, ...){
183     if (class(X) %in% c('integer', 'numeric')){
184       if (!is.null(GRPS)){
185         if (is.null(FUN)){
186           outLine <- c(paste('$', formatC(tapply(X, GRPS, mean, na.rm = T), format = 'f', ...), '\\; (',
187             formatC(tapply(X, GRPS, sd, na.rm = T), format = 'f', ...), ')$', sep = '''),
188             paste('$', formatC(mean(X, na.rm = T), format = 'f', ...), '\\; (',
189               formatC(sd(X, na.rm = T), format = 'f', ...), ')$', sep = '''))
190           if (overall == FALSE) outLine <- outLine[-(length(unique(GRPS)) + 1)]
191         }
192         if (do.test){
193           if (is.null(TEST)){
194             outLine <- c(outLine, pvalFormat(t.test(X ~ GRPS)$p.value))
195           } else {
196             if (is.null(ARG[[1]])){
197               outLine <- c(outLine, pvalFormat(do.call(TEST, list(as.formula(X ~ GRPS))$p.value))
198             } else {
199               ARG <- c(X ~ GRPS, ARG); names(ARG)[1] <- 'formula'
200               outLine <- c(outLine, pvalFormat(do.call(TEST, ARG)$p.value))
201             }
202           }
203         }
204       } else {
205         if (is.null(FUN)){
206           outLine <- paste('$', formatC(mean(X), format = 'f', ...), '\\; (',
207             formatC(sd(X), format = 'f', ...), ')$', sep = ''')
208         }
209       }
210     }
211     if (class(X) %in% c('character', 'factor')){
212       p.test <- NULL
213       if (is.null(GRPS)){
214         if (is.null(FUN)){
215           outLine <- suppressWarnings(paste(table(X), '\\; (',
216             formatC(prop.table(table(X)) * 100, digits = 1, format = 'f'), '\\%', ')',
217             sep = ''))
218           outLine <- as.matrix(paste('$', outLine, '$', sep = ''), ncol = 1)
219         }
220       } else {
221         if (is.null(FUN)){
222           outLine <- paste('$', table(X, GRPS), '\\; (',
223             suppressWarnings(formatC(prop.table(table(X, GRPS), margin = 2) * 100,
224               digits = 1, format = 'f')), '\\%', ')$', sep = ''')
225           outLine <- cbind(matrix(outLine, ncol = length(unique(GRPS))),
226             paste('$', table(X), '\\; (',
227               suppressWarnings(formatC(prop.table(table(X)) * 100, digits = 1, format = 'f')),
228               '\\%', ')$', sep = ''))

```

```

228     if(overall == FALSE) outLine <- outLine[, -ncol(outLine)]
229   }
230   if(do.test & nrow(outLine) != 1){
231     if(is.null(TEST)){
232       p.test <- c(pvalFormat(chisq.test(table(X, GRPS), correct = FALSE)$p.value), rep('', nrow(outLine)))
233     } else{
234       if(is.null(ARG[[1]])){
235         p.test <- c(pvalFormat(do.call(TEST, list(table(X, GRPS))$p.value), rep('', nrow(outLine)))
236       } else{
237         ARG$x <- table(X, GRPS)
238         p.test <- c(pvalFormat(do.call(TEST, ARG)$p.value), rep('', nrow(outLine)))
239       }
240     }
241   } else{
242     p.test <- c('-', '')
243   }
244 }
245 outLine <- rbind(rep('', ncol(outLine)), outLine)
246 if(!is.null(p.test)) outLine <- cbind(outLine, p.test)
247 }
248 return(outLine)
249 }
250
251 if(!is.null(group)){
252   if(class(group) %in% c('factor', 'character')) GROUPS <- which(names(DATA) == group)
253   if(class(group) == 'integer') GROUPS <- group
254 } else{GROUPS <- NULL}
255
256 if(!class(vars) %in% c('formula', 'character', 'integer')) stop('Unsupported class for vars argument.')
257
258 if(class(vars) == 'formula'){
259   GROUPS <- which(names(DATA) == as.character(vars[2]))
260   VAR_NAMES <- strsplit(as.character(vars[3]), '+', fixed = TRUE)[[1]]
261   WHICH <- numeric(length(VAR_NAMES))
262   for(ii in 1:length(WHICH)) WHICH[ii] <- which(names(DATA) == VAR_NAMES[ii])
263 }
264
265 if(class(vars) == 'character') WHICH <- which(names(DATA) %in% vars)
266 if(class(vars) == 'integer') WHICH <- vars
267
268 if(!is.null(GROUPS)) GRPS <- DATA[, GROUPS] else{GRPS <- NULL}
269
270 out <- NULL
271 R_NAMES <- NULL
272 if(length(do.test) == 1) do.test <- rep(do.test, length(WHICH))
273 for(i in 1:length(WHICH)){
274   out <- rbind(out, makeTab(DATA[, WHICH[i]], GRPS, do.test[i], tests[i], FUNS[i, ], ARG = arg[[i]], ...))
275   if(class(DATA[, WHICH[i]]) %in% c('integer', 'numeric')) R_NAMES <- c(R_NAMES, names(DATA)[WHICH[i]])
276   if(class(DATA[, WHICH[i]]) %in% c('character', 'factor')){
277     R_NAMES <- c(R_NAMES, names(DATA)[WHICH[i]], paste('~~~~', names(table(DATA[, WHICH[i]])), sep = ''))
278   }
279 }
280 rownames(out) <- R_NAMES
281 if(is.null(GROUPS)) colnames(out) <- paste('Total ($n=', nrow(DATA), '$)', sep = '')
282 if(!is.null(GROUPS)){
283   if(overall == TRUE){
284     colnames(out) <- c(paste(sort(unique(GRPS)), ' ($n=', table(GRPS), '$)', sep = ''), paste('Total ($n=', nrow(
285       DATA), '$)', sep = ''),
286       'p-value')
287   } else{colnames(out) <- c(paste(sort(unique(GRPS)), ' ($n=', table(GRPS), '$)', sep = ''), 'p-value')}
288 }
289 return(out)
290 }
291
292 ###import data
293 dat = read.csv('82682.csv')
294 newdat = dat[order(dat$EMPLOYEE_ID),]
295 #newdat$HomeDept = gsub(" $", "", newdat$HomeDept, perl=T)
296 newdat$HomeDept = str_trim(newdat$HomeDept)
297 #modify the format of the salary
298 for (i in 8:18){
299   newdat[, i] = as.numeric(gsub(",","", newdat[, i]))
300 }
301 for (i in c(1:7, 19:20, 23:24, 28:34)){
302   newdat[, i] = factor(newdat[, i])
303 }
304 #clean up the empty levels of the degree and homedepth
305 for (i in 1:length(newdat$Degree)){
306   if (newdat$Degree[i] == "" | newdat$Degree[i] == " "){newdat$Degree[i] = NA}

```

```

306 }
307 newdat$Degree = droplevels(newdat$Degree)
308 for (i in 1:length(newdat$HomeDept)){
309   if (newdat$HomeDept[i] == "MEDICINE" ) {newdat$HomeDept[i] = "MEDICINE"}
310   if (newdat$HomeDept[i] == "PSYCHIATRY" ) {newdat$HomeDept[i] = "PSYCHIATRY"}
311   if (newdat$HomeDept[i] == "FAMILY & PREVENTIVE MEDICINE" ) {newdat$HomeDept[i] = "FAMILY & PREVENTIVE MEDICINE"}
312 }
313 newdat$HomeDept = droplevels(newdat$HomeDept)
314 summary(newdat$HomeDept)
315
316 #for (i in 1:length(newdat$Rank)){
317 # if (newdat$Rank[i] == "" | newdat$Rank[i] == " ") {newdat$Rank[i] = NA}
318 #}
319 #newdat$Rank = droplevels(newdat$Rank)
320 #for (i in 1:length(newdat$Series)){
321 # if (newdat$Series[i] == "Adj"){newdat$Series[i] = "ADJ"}
322 # else if (newdat$Series[i] == "Clin"){newdat$Series[i] = "CLIN"}
323 # else if (newdat$Series[i] == "Clin/X"){newdat$Series[i] = "CLINx"}
324 # else if (newdat$Series[i] == "Non"){newdat$Series[i] = ""}
325 #}
326 #newdat$Series = droplevels(newdat$Series)
327
328 #create continous variable of fiscal year just in case we need to use it later
329 year = NULL
330 for (i in 1:length(newdat$FiscalYear)){
331   if (newdat$FiscalYear[i] == "2009-2010"){year[i] = 2010}
332   else if (newdat$FiscalYear[i] == "2010-2011"){year[i] = 2011}
333   else if (newdat$FiscalYear[i] == "2011-2012"){year[i] = 2012}
334   else if (newdat$FiscalYear[i] == "2012-2013"){year[i] = 2013}
335   else if (newdat$FiscalYear[i] == "2013-2014"){year[i] = 2014}
336 }
337
338 #we use it in the next testing part but not necessary
339 a = c(21, 22, 25, 26, 35:37)
340 new = matrix(rep(NA, nrow(newdat)*7), ncol = 7)
341 for (i in 1:7){
342   new[, i] = abs(as.numeric(difftime(as.Date(newdat[, a[i]], format="%m%d/%y"), "2014-11-04", units = "days"))/365)
343 }
344 newdat = cbind(newdat, new)
345 names(newdat)[38:44] = c("original_hire", "seperation", "first_appt", "hire", "md_degree", "phd_degree", "other_degree")
346
347 #create the variable of years since degree
348 degree_year = NULL
349 for (i in 1:length(newdat$FiscalYear)){
350   degree_year[i] = max(newdat$md_degree[i], newdat$phd_degree[i], newdat$other_degree[i], na.rm = T)
351   if (degree_year[i] == -Inf){degree_year[i] = NA}
352 }
353 newdat = cbind(newdat, degree_year)
354 names(newdat)[45] = "degree_year"
355 newdat = cbind(newdat, year)
356 names(newdat)[46] = "year"
357
358 ###testing
359 test = newdat[, c(2, 3, 21, 25, 26, 28, 29, 35:37, 45)]
360 test.unique = unique(test)
361 test.unique = test.unique[order(test.unique$EMPLOYEE_ID),]
362 id = test.unique$EMPLOYEE_ID[duplicated(test.unique$EMPLOYEE_ID)]
363 test.all = test.unique[which(test.unique$EMPLOYEE_ID %in% id),]
364 test.all2 = newdat[which(newdat$original_hire < newdat$first_appt), c(2, 21, 25)]
365 test.all2 = unique(test.all2) ###test.all2 should be empty if data is consistent across years
366
367 ###update the data
368 #modify all the date variable to years till the year they are recorded
369 a = c(21, 22, 25, 26, 35:37)
370 new = matrix(rep(NA, nrow(newdat)*7), ncol = 7)
371 for (i in 1:7){
372   for (j in 1:length(newdat$FiscalYear)){
373     if (newdat[j, 46] == 2010){
374       new[j, i] = abs(as.numeric(difftime(as.Date(newdat[j, a[i]], format="%m%d/%y"), "2010-12-31", units = "days"))
375         /365)
376     }
377     else if (newdat[j, 46] == 2011){
378       new[j, i] = abs(as.numeric(difftime(as.Date(newdat[j, a[i]], format="%m%d/%y"), "2011-12-31", units = "days"))
379         /365)
380     }
381     else if (newdat[j, 46] == 2012){
382       new[j, i] = abs(as.numeric(difftime(as.Date(newdat[j, a[i]], format="%m%d/%y"), "2012-12-31", units = "days"))
383         /365)
384     }
385   }
386 }

```

```

381   }
382   else if (newdat[j, 46] == 2013){
383     new[j, i] = abs(as.numeric(difftime(as.Date(newdat[j, a[i]], format="%m/%d/%y"), "2013-12-31", units = "days"))
384                   /365)
385   }
386   else if (newdat[j, 46] == 2014){
387     new[j, i] = abs(as.numeric(difftime(as.Date(newdat[j, a[i]], format="%m/%d/%y"), "2014-12-31", units = "days"))
388                   /365)
389   }
390 }
391 #modify years since firt_appt
392 newdat[, 38:44] = new
393 #for (i in 1:length(newdat$FiscalYear)){
394 #  newdat$first_appt[i] = ifelse(newdat$first_appt[i] <= newdat$original_hire[i], newdat$first_appt[i],
395 #                               newdat$original_hire[i] )
396 #}
397 for (i in 1:length(newdat$FiscalYear)){
398   newdat$first_appt[i] = newdat$original_hire[i]
399 }
400 #create two new columns represent the centered years since degree and years of experience at ucsd
401 newdat = cbind(newdat, newdat$first_appt-mean(newdat$first_appt, na.rm=T), newdat$degree_year-mean(newdat$degree_year,
402   na.rm=T))
403 names(newdat)[47:48] = c("c_first_appt", "c_degree_year")
404 #create adjusted salary
405 a = newdat[, 16]/newdat[, 17]
406 newdat = cbind(newdat, a)
407 names(newdat)[49] = "adj.salary"
408 #relevel rank and series
409 newdat$Gender <- relevel(newdat$Gender, ref="M")
410 newdat$Rank <- relevel(newdat$Rank, ref="Asst Prof")
411 newdat$Series <- relevel(newdat$Series, ref="FTE")
412 #exclude those whose appt% < 0.5
413 newdat = newdat[which(newdat$Calc.Appt.. >= 0.5),]
414 names(newdat)[34] = "URM"
415 newdat[which(newdat$URM == "Unknown"), "URM"] = rep("", length(newdat[which(newdat$URM == "Unknown"), 1]))
416 newdat$URM = droplevels(newdat$URM)
417 newdat$URM <- ifelse(newdat$URM=="URM", 1,0)
418 newdat$URM = factor(newdat$URM)
419 newdat$URM = revalue(newdat$URM, c("1" ="URM", "0"="Not URM"))
420
421 ###correlation between rank and years since degree or years of experience at UCSD
422 bsa = function(fiscalyear, experience){
423   data = newdat[which(newdat$year == fiscalyear), c("Rank", "experience")]
424   data = na.omit(data)
425   x = data[which(data$Rank == "Asst Prof"), "experience"]
426   y = data[which(data$Rank == "Assoc Prof"), "experience"]
427   z = data[which(data$Rank == "Professor"), "experience"]
428   n1 = length(x)
429   n2 = length(y)
430   n3 = length(z)
431   sum = 0
432   for (i in 1:n1){
433     for (j in 1:n2){
434       for (k in 1:n3){
435         obs_rank = order(c(x[i], y[j], z[k]))
436         dist = (obs_rank[1] - 1)^2 + (obs_rank[2] - 2)^2 + (obs_rank[3] - 3)^2
437         sum = sum + dist
438       }
439     }
440   }
441   c = (3^3 - 3)/6
442   p = 1 - sum/(n1 * n2 * n3)/c
443   return(p)
444 }
445 #results
446 #> bsa(2011, "degree_year")
447 #> bsa(2012, "degree_year")
448 #> bsa(2013, "degree_year")
449 #> bsa(2014, "degree_year")
450 #> bsa(2011, "degree_year")
451 #> bsa(2012, "degree_year")
452 #> bsa(2013, "degree_year")
453 #> bsa(2014, "degree_year")
454
455 ###contingency tables
456 subgroup.categorical = function(data, Var, group, outcome, total){
457   data = data

```

```

457 subgroup.name = levels(factor(data[, Var]))
458 subgroup.name1 = levels(factor(data[, group]))
459 if (total == T){
460   sub.data = aggregate(as.formula(paste(outcome, "~", Var, "+", group, sep = "")), data = newdat, median)
461   sub.data2 = aggregate(as.formula(paste(outcome, "~", Var, sep = "")), data = newdat, median)
462   out = matrix(sub.data[, outcome], nrow = length(subgroup.name), byrow = F)
463   out2 = matrix(sub.data2[, outcome], nrow = length(subgroup.name), byrow = F)
464   out = cbind(out2, out)
465   rownames(out) = subgroup.name
466   colnames(out) = c("Total", subgroup.name1)
467   out = prettyNum(formatC(out ,digits = 0, format = 'f'), big.mark = ",")
468 }
469 else{
470   sub.data = aggregate(as.formula(paste(outcome, "~", Var, "+", group, sep = "")), data = newdat, median)
471   out = matrix(sub.data[, outcome], nrow = length(subgroup.name), byrow = F)
472   rownames(out) = subgroup.name
473   colnames(out) = subgroup.name1
474   out = prettyNum(formatC(out ,digits = 0, format = 'f'), big.mark = ",")
475 }
476 return(out)
477 }
478
479 ###descriptive tables
480 dat9 = newdat[which(newdat$FiscalYear == "2009-2010"), ]
481 dat9$HomeDept = droplevels(dat9$HomeDept)
482 dat10 = newdat[which(newdat$FiscalYear == "2010-2011"), ]
483 dat10$HomeDept = droplevels(dat10$HomeDept)
484 dat11 = newdat[which(newdat$FiscalYear == "2011-2012"), ]
485 dat11$HomeDept = droplevels(dat11$HomeDept)
486 dat12 = newdat[which(newdat$FiscalYear == "2012-2013"), ]
487 dat12$HomeDept = droplevels(dat12$HomeDept)
488 dat13 = newdat[which(newdat$FiscalYear == "2013-2014"), ]
489 dat13$HomeDept = droplevels(dat13$HomeDept)
490
491 destab = function(data){
492   dat9 = data
493   tab9 = demoTab(Gender ~ Degree + degree_year + Rank + first_appt + Series + HomeDept + URM, DATA = dat9, digits = 2)
494   p = rep("", nrow(tab9))
495   p[ c(1, 11, 12, 16, 17, 23, 44)] = c(pvalFormat(pvalue(chisq_test(Gender~Degree, data=dat9))), tab9[11, 4],
496     pvalFormat(pvalue(chisq_test(Gender~Rank, data=dat9))), tab9[16, 4]
497     , pvalFormat(pvalue(chisq_test(Gender~Series, data=dat9))), pvalFormat(pvalue(chisq_test(
498       Gender~HomeDept, data=dat9))), pvalFormat(pvalue(chisq_test(Gender~URM, data=dat9)))
499     )
500   tab9 = cbind(tab9[, -4], p)
501   colnames(tab9)[4] = 'p-value'
502   missing = rep("", nrow(tab9))
503   missing[ c(1, 11, 12, 16, 17, 23, 44)] = c(length(which(is.na(dat9$Degree))), length(which(is.na(dat9$degree_year))),
504     length(which(is.na(dat9$Rank))), length(which(is.na(dat9$first_appt))), length(which(is.na(dat9$Series))),
505     length(which(is.na(dat9$HomeDept))), length(which(is.na(dat9$URM))))
506   tab9 = cbind(tab9, missing)
507   colnames(tab9)[5] = 'Missing'
508   rownames(tab9) = c("\\bfseries Degree", "~~~~DO", "~~~~DO/PhD", "~~~~DPM", "~~~~MD", "~~~~MD/PhD", "~~~~NonDoc",
509     "~~~~OtherDoc", "~~~~PharmD", "~~~~PhD", "\\bfseries Years since degree", "\\bfseries Rank", "~~~~Asst Prof",
510     "~~~~Assoc Prof", "~~~~Professor", "\\bfseries Years since employment at UCSD", "\\bfseries Series", "~~~~FTE",
511     "~~~~ADJ", "~~~~CLIN", "~~~~CLINx", "~~~~IR", "\\bfseries HomeDept", "~~~~ANESTHESIOLOGY", "~~~~BIOENGINEERING",
512     "~~~~CELL \\& MOL MED PROG", "~~~~CHEMISTRY \\& BIOCHEMISTRY", "~~~~DEP. OF REPRODUCTIVE MEDICINE", "~~~~
513     DIVISION OF BIOLOGICAL SCI.", "~~~~FAMILY \\& PREVENTIVE MEDICINE", "~~~~MEDICINE", "~~~~MEDSCH/EMERG MED SVC",
514     "~~~~NEUROSCIENCES", "~~~~OPHTHALMOLOGY", "~~~~ORTHOPAEDIC SURGERY", "~~~~PATHOLOGY", "~~~~PEDIATRICS", "~~~~
515     PHARMACOLGY", "~~~~PSYCHIATRY", "~~~~RADIATION MED \\& APPLIED SCI", "~~~~RADIOLOGY", "~~~~SCH OF PHARMACY AND
516     PHARM. SCI", "~~~~SURGERY", "\\bfseries URM", "~~~~Not URM", "~~~~URM")
517   return(tab9)
518 }
519
520 ###scatter plots
521 d1 = newdat[which(newdat$first_appt <= 10),]
522 d2 = newdat[which(newdat$first_appt <= 20 & newdat$first_appt > 10),]
523 d3 = newdat[which(newdat$first_appt <= 30 & newdat$first_appt > 20),]
524 d4 = newdat[which(newdat$first_appt > 30),]
525 d1a = floor(d1$first_appt)
526 d1 = cbind(d1, d1a)
527 d1$d1a = factor(d1$d1a)
528 d2a = floor(d2$first_appt)
529 d2 = cbind(d2, d2a)
530 d2$d2a = factor(d2$d2a)
531 d3a = floor(d3$first_appt)
532 d3 = cbind(d3, d3a)
533 d3$d3a = factor(d3$d3a)
534 d4a = floor(d4$first_appt)
535 d4 = cbind(d4, d4a)

```



```

523
524 d4$d4a = factor(d4$d4a)
525 dy = NULL
526 for (i in 1:nrow(newdat)){
527   if (!is.na(newdat$degree_year[i])){
528     if (newdat$degree_year[i] <= 10){
529       dy[i] = 0
530     }
531     else if (newdat$degree_year[i] <= 20 & newdat$degree_year[i] > 10){
532       dy[i] = 1
533     }
534     else if (newdat$degree_year[i] <= 30 & newdat$degree_year[i] > 20){
535       dy[i] = 2
536     }
537     else {
538       dy[i] = 3
539     }
540   }
541   else {
542     dy[i] = NA
543   }
544 }
545 dy = factor(dy)
546 newdat = cbind(newdat, dy)
547
548 fmt <- function(){
549   function(x) format(x, nsmall = -5, scientific = FALSE)
550 }
551
552 pdf(paste('scatplot1', '.pdf', sep = ''), height = 5, width = 7)
553 ggplot(newdat[which(!is.na(newdat$Gender)),], aes(x=degree_year, y=adj.salary, group = Gender, color = Gender)) +
554   geom_point(shape=1, alpha=0.7) + geom_smooth() + facet_grid(. ~ year) + labs(x = "Years Since Degree", y = "
555     Total Salary") + scale_y_log10(labels = dollar, breaks = seq(100000, 1000000, by = 300000)) + scale_x_continuous(breaks = seq(10,55,by = 20)) + scale_colour_tableau()
554 dev.off()
555
556 pdf(paste('scatplot2', '.pdf', sep = ''), height = 5, width = 7)
557 ggplot(newdat[which(!is.na(newdat$URM)),], aes(x=degree_year, y=adj.salary, group = URM, color = URM)) + geom_point(
558   shape=1, alpha=0.7) + geom_smooth(size=0.6) + facet_grid(. ~ year) + labs(x = "Years Since Degree", y = "Total
559     Salary") + scale_y_log10(labels = dollar, breaks = seq(100000, 1000000, by = 300000)) + scale_x_continuous(breaks
560     = seq(10,55,by = 20)) + scale_colour_tableau()
562 dev.off()
563
564 pdf(paste('scatplot3', '.pdf', sep = ''), height = 6, width = 8)
565 ggplot(newdat, aes(x=first_appt, y=adj.salary)) + geom_point(shape=1, alpha=0.7) + geom_smooth() + facet_grid(year ~
566   .) + labs(x = "Years of Experience at UCSD", y = "Total Salary") + scale_y_log10(labels = dollar, breaks = seq
567     (100000, 1000000, by = 300000))
562 dev.off()
563
564 pdf(paste('scatplot5', '.pdf', sep = ''), height = 5, width = 7)
565 ggplot(newdat, aes(x=first_appt, y=adj.salary, group=Gender, color=Gender)) + geom_point(shape=1, alpha=0.7) +
566   geom_smooth(size = 0.6) + facet_grid(year ~ .) + labs(x = "Years of Experience at UCSD", y = "Total Salary") +
567   scale_y_log10(labels = dollar, breaks = seq(100000, 1000000, by = 300000)) + scale_colour_tableau()
566 dev.off()
567
568 pdf(paste('scatplot6', '.pdf', sep = ''), height = 5, width = 7)
569 ggplot(newdat, aes(x=first_appt, y=adj.salary, group=Rank, color=Rank)) + geom_point(shape=1, alpha=0.7) + geom_smooth
570   (size = 0.6) + facet_grid(year ~ .) + labs(x = "Years of Experience at UCSD", y = "Total Salary") +
571   scale_y_log10(labels = dollar, breaks = seq(100000, 1000000, by = 300000))
570 dev.off()
571
572 pdf(paste('scatplot7', '.pdf', sep = ''), height = 6, width = 9)
573 ggplot(newdat, aes(x=first_appt, y=adj.salary, group=interaction(Rank, Gender), color=interaction(Rank, Gender))) +
574   geom_point(shape=1, alpha=0.7) + geom_smooth(size=0.6) + facet_grid(year ~ .) + labs(x = "Years of Experience at
575     UCSD", y = "Total Salary") + scale_y_log10(labels = dollar, breaks = seq(100000, 1000000, by = 300000))
574 dev.off()
575
576 pdf(paste('scatplot4', '.pdf', sep = ''), height = 5, width = 7)
577 ggplot(newdat[which(!is.na(newdat$URM)),], aes(x=first_appt, y=adj.salary, group = URM, color = URM)) + geom_point(
578   shape=1, alpha=0.7) + geom_smooth(size=0.7) + facet_grid(. ~ year) + labs(x = "Years of Experience at UCSD", y
579     = "Total Salary") + scale_y_log10(labels = dollar, breaks = seq(100000, 1000000, by = 300000)) +
580   scale_x_continuous(breaks = seq(10,55,by = 20)) + scale_colour_tableau()
578 dev.off()
579
580
581
582 ###spaghetti plots
583 b = newdat[which(!is.na(newdat$Gender)),]
584 pdf(paste('plot1', '.pdf', sep = ''), height = 4, width = 6)

```



```

585 ggplot(data = newdat[which(!is.na(newdat$Gender)),], aes(x = year, y = adj.salary, group = EMPLOYEE_ID, color = Gender
) + geom_line(alpha = 1) + labs(x = "Fiscal Year", y = "Total Salary") + scale_y_log10(labels = dollar, breaks =
seq(100000, 1000000, by = 300000))
586 dev.off()
587
588 b = newdat[which(!is.na(newdat$URM)),]
589 pdf(paste('plot2', '.pdf', sep = ''), height = 4, width = 6)
590 cols <- c("LINE1"="bisque", "LINE2"="black")
591 ggplot(data = newdat[which(newdat$URM == "Not URM"),], aes(x = year, y = adj.salary, group = EMPLOYEE_ID, colour = "
LINE1") + geom_line(alpha = 1) + geom_line(data = newdat[which(newdat$URM == "URM"),], aes(x=year, y=adj.salary,
group = EMPLOYEE_ID, colour = "LINE2")) + labs(x = "Fiscal Year", y = "Total Salary") + scale_y_log10(labels =
dollar, breaks = seq(100000, 1000000, by = 300000)) + scale_colour_manual(values=cols, name = "", labels = c("Not
URM", "URM"))
592 dev.off()
593
594
595
596 ### fit model
597 newdat2 = newdat[, c(1:5, 24, 28, 34, 46:47, 49)]
598 newdat2$HomeDept = droplevels(newdat2$HomeDept)
599 fit3.0 = lmer(log(adj.salary) ~ Rank + Gender + Series + HomeDept + c_first_appt + year + Degree + I(c_first_appt^2) +
(1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
600 temp1 = lme(log(adj.salary) ~ Rank + Gender + Series + HomeDept + c_first_appt + year + Degree + I(c_first_appt^2),
random = ~ 1|EMPLOYEE_ID, data = na.omit(newdat2))
601 temp00 = lme(log(adj.salary) ~ 1, random = ~ 1|EMPLOYEE_ID, data = na.omit(newdat2))
602 fit3.1 = lmer(log(adj.salary) ~ Gender + Series + HomeDept + c_first_appt + year + Degree + I(c_first_appt^2) + (1|
EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
603 fit3.2 = lmer(log(adj.salary) ~ Rank + Series + HomeDept + c_first_appt + year + Degree + I(c_first_appt^2) + (1|
EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
604 fit3.3 = lmer(log(adj.salary) ~ Rank + Gender + HomeDept + c_first_appt + year + Degree + I(c_first_appt^2) + (1|
EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
605 fit3.4 = lmer(log(adj.salary) ~ Rank + Gender + Series + c_first_appt + year + Degree + I(c_first_appt^2) + (1|
EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
606 fit3.5 = lmer(log(adj.salary) ~ Rank + Gender + Series + HomeDept + year + Degree + I(c_first_appt^2) + (1|EMPLOYEE_ID
), data = na.omit(newdat2), REML = F)
607 fit3.6 = lmer(log(adj.salary) ~ Rank + Gender + Series + HomeDept + c_first_appt + Degree + I(c_first_appt^2) + (1|
EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
608 fit3.7 = lmer(log(adj.salary) ~ Rank + Gender + Series + HomeDept + c_first_appt + year + I(c_first_appt^2) + (1|
EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
609 fit3.8 = lmer(log(adj.salary) ~ Rank + Gender + Series + HomeDept + c_first_appt + year + Degree + (1|EMPLOYEE_ID),
data = na.omit(newdat2), REML = F)
610 like.test = NULL
611 like.test[2] = anova(fit3.0, fit3.1)$Pr[2]
612 like.test[5] = anova(fit3.0, fit3.2)$Pr[2]
613 like.test[7] = anova(fit3.0, fit3.3)$Pr[2]
614 like.test[12] = anova(fit3.0, fit3.4)$Pr[2]
615 like.test[32] = anova(fit3.0, fit3.5)$Pr[2]
616 like.test[33] = anova(fit3.0, fit3.6)$Pr[2]
617 like.test[34] = anova(fit3.0, fit3.7)$Pr[2]
618 like.test[43] = anova(fit3.0, fit3.8)$Pr[2]
619 sum.tab = cbind(summary(fit3.0)$coefficients,
620 pvalFormat(2 * pnorm(abs(summary(fit3.0)$coefficients[, 3]), lower.tail = FALSE)))
621 sum.tab = rbind(sum.tab[1,], rep(NA, 4), sum.tab[2:3,], rep(NA, 4), sum.tab[4,], rep(NA, 4), sum.tab[5:8,], rep(NA, 4)
, sum.tab[9:29,], rep(NA, 4), sum.tab[30:38,])
622 sum.tab = cbind(sum.tab, like.test)
623 sum.tab = matrix(sum.tab, ncol = 5)
624 rownames(sum.tab) = c("(Intercept)", "\\bfseries Rank", "~~~~Assoc Prof", "~~~~Professor", "\\bfseries Gender", "~~~~
Female", "\\bfseries Series", "~~~~ADJ", "~~~~CLIN", "~~~~CLINx", "~~~~IR", "\\bfseries Home Department", "~~~~
BIOENGINEERING", "~~~~CELL \\& MOL MED PROG", "~~~~CHEMISTRY \\& BIOCHEMISTRY", "~~~~DEP. OF REPRODUCTIVE
MEDICINE", "~~~~DIVISION OF BIOLOGICAL SCI.", "~~~~FAMILY \\& PREVENTIVE MEDICINE", "~~~~MEDICINE", "~~~~MEDSCH/
EMERG MED SVC", "~~~~NEUROSCIENCES", "~~~~OPHTHALMOLOGY", "~~~~ORTHOPAEDIC SURGERY", "~~~~PATHOLOGY", "~~~~
PEDIATRICS", "~~~~PHARMACOLOGY", "~~~~PSYCHIATRY", "~~~~RADIATION MED \\& APPLIED SCI", "~~~~RADIOLOGY", "~~~~SCH
OF PHARMACY AND PHARM. SCI", "~~~~SURGERY", "\\bfseries Years of experience at UCSD", "\\bfseries Fiscal Year",
"\\bfseries Degree", "~~~~DO/PhD", "~~~~DPM", "~~~~MD", "~~~~MD/PhD", "~~~~NonDoc", "~~~~OtherDoc", "~~~~PharmD",
"~~~~PhD", "\\bfseries Years of experience at UCSD^{2}$")
625 colnames(sum.tab) = c("Coefficient", "Std Error", "Z-Value", "Z-Test", "LRT")
626 sum.tab[, c(1, 2)] = paste('$', formatC(as.numeric(sum.tab[, c(1, 2)]), digits = 4, format = 'f'), '$', sep = '')
627 sum.tab[, c(3)] = paste('$', formatC(as.numeric(sum.tab[, c(3)]), digits = 2, format = 'f'), '$', sep = '')
628 sum.tab[, c(5)] = pvalFormat(sum.tab[, c(5)])
629 sum.tab[c(1, 3:4, 6, 8:11, 13:31, 35:42), 5] = rep('', 35)
630 sum.tab[c(2, 5, 7, 12, 34), 1:4] = rep('', 20)
631 pdf(paste('residual1', '.pdf', sep = ''), height = 5, width = 6)
632 plot(fitted(fit3.0), scale(residuals(fit3.0)), col="dodgerblue", xlab = "Fitted Values", ylab = "Standardized
Conditional Residuals")
633 abline(h=0)
634 dev.off()
635
636
637 fit3.02 = lmer(log(adj.salary) ~ Rank + URM + Series + HomeDept + c_first_appt + year + Degree + I(c_first_appt^2) +

```

```

(1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
638 temp2 = lme(log(adj.salary) ~ Rank + URM + Series + HomeDept + c_first_appt + year + Degree + I(c_first_appt^2),
  random = ~ 1|EMPLOYEE_ID, data = na.omit(newdat2))
639 fit3.12 = lmer(log(adj.salary) ~ URM + Series + HomeDept + c_first_appt + year + Degree + I(c_first_appt^2) + (1|
  EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
640 fit3.22 = lmer(log(adj.salary) ~ Rank + Series + HomeDept + c_first_appt + year + Degree + I(c_first_appt^2) + (1|
  EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
641 fit3.32 = lmer(log(adj.salary) ~ Rank + URM + HomeDept + c_first_appt + year + Degree + I(c_first_appt^2) + (1|
  EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
642 fit3.42 = lmer(log(adj.salary) ~ Rank + URM + Series + c_first_appt + year + Degree + I(c_first_appt^2) + (1|
  EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
643 fit3.52 = lmer(log(adj.salary) ~ Rank + URM + Series + HomeDept + year + Degree + I(c_first_appt^2) + (1|EMPLOYEE_ID),
  data = na.omit(newdat2), REML = F)
644 fit3.62 = lmer(log(adj.salary) ~ Rank + URM + Series + HomeDept + c_first_appt + Degree + I(c_first_appt^2) + (1|
  EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
645 fit3.72 = lmer(log(adj.salary) ~ Rank + URM + Series + HomeDept + c_first_appt + year + I(c_first_appt^2) + (1|
  EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
646 fit3.82 = lmer(log(adj.salary) ~ Rank + URM + Series + HomeDept + c_first_appt + year + Degree + (1|EMPLOYEE_ID), data
  = na.omit(newdat2), REML = F)
647 like.test2 = NULL
648 like.test2[2] = anova(fit3.02, fit3.12)$Pr[2]
649 like.test2[5] = anova(fit3.02, fit3.22)$Pr[2]
650 like.test2[7] = anova(fit3.02, fit3.32)$Pr[2]
651 like.test2[12] = anova(fit3.02, fit3.42)$Pr[2]
652 like.test2[32] = anova(fit3.02, fit3.52)$Pr[2]
653 like.test2[33] = anova(fit3.02, fit3.62)$Pr[2]
654 like.test2[34] = anova(fit3.02, fit3.72)$Pr[2]
655 like.test2[43] = anova(fit3.02, fit3.82)$Pr[2]
656 sum.tab2 = cbind(summary(fit3.02)$coefficients,
  pvalFormat(2 * pnorm(abs(summary(fit3.02)$coefficients[, 3]), lower.tail = FALSE)))
658 sum.tab2 = rbind(sum.tab2[1,], rep(NA, 4), sum.tab2[2:3,], rep(NA, 4), sum.tab2[4,], rep(NA, 4), sum.tab2[5:8,], rep(
  NA, 4), sum.tab2[9:29,], rep(NA, 4), sum.tab2[30:38,])
659 sum.tab2 = cbind(sum.tab2, like.test2)
660 sum.tab2 = matrix(sum.tab2, ncol = 5)
661 rownames(sum.tab2) = c("Intercept", "\\bfseries Rank", "~~~~Assoc Prof", "~~~~Professor", "\\bfseries URM", "~~~~URM
  ", "\\bfseries Series", "~~~~ADJ", "~~~~CLIN", "~~~~CLINx", "~~~~IR", "\\bfseries Home Department", "~~~~
  BIOENGINEERING", "~~~~CELL \\& MOL MED PROG", "~~~~CHEMISTRY \\& BIOCHEMISTRY", "~~~~DEP. OF REPRODUCTIVE
  MEDICINE", "~~~~DIVISION OF BIOLOGICAL SCI.", "~~~~FAMILY \\& PREVENTIVE MEDICINE", "~~~~MEDICINE", "~~~~MEDSCH/
  EMERG MED SVC", "~~~~NEUROSCIENCES", "~~~~OPHTHALMOLOGY", "~~~~ORTHOPAEDIC SURGERY", "~~~~PATHOLOGY", "~~~~
  PEDIATRICS", "~~~~PHARMACOLOGY", "~~~~PSYCHIATRY", "~~~~RADIATION MED \\& APPLIED SCI", "~~~~RADIOLOGY", "~~~~SCH
  OF PHARMACY AND PHARM. SCI", "~~~~SURGERY", "\\bfseries Years of experience at UCSD", "\\bfseries Fiscal Year",
  "\\bfseries Degree", "~~~~DO/PhD", "~~~~DPM", "~~~~MD", "~~~~MD/PhD", "~~~~NonDoc", "~~~~OtherDoc", "~~~~PharmD",
  "~~~~PhD", "\\bfseries Years of experience at UCSD${2}$")
662 colnames(sum.tab2) = c("Coefficient", "Std Error", "Z-Value", "Z-Test", "LRT")
663 sum.tab2[, c(1, 2)] = paste('$', formatC(as.numeric(sum.tab2[, c(1, 2)]), digits = 4, format = 'f'), '$', sep = '')
664 sum.tab2[, c(3)] = paste('$', formatC(as.numeric(sum.tab2[, c(3)]), digits = 2, format = 'f'), '$', sep = '')
665 sum.tab2[, c(5)] = pvalFormat(sum.tab2[, c(5)])
666 sum.tab2[, c(1, 3:4, 6, 8:11, 13:31, 35:42), 5] = rep('', 35)
667 sum.tab2[, c(2, 5, 7, 12, 34), 1:4] = rep('', 20)
668 pdf(paste('residual2', '.pdf', sep = ''), height = 5, width = 6)
669 plot(fitted(fit3.02), scale(residuals(fit3.02)), col="dodgerblue", xlab = "Fitted Values", ylab = "Standardized
  Conditional Residuals")
670 abline(h=0)
671 dev.off()
672
673 sum = rbind(sum.tab[5:6, ], sum.tab2[5:6, ])
674 sum[2, 5] = sum[1, 5]
675 sum[4, 5] = sum[3, 5]
676 sum = sum[, c(2, 4), ]
677 rownames(sum) = c("Gender(female)", "URM(URM)")
678
679 fit3.03 = lmer(log(adj.salary) ~ Rank + Gender + URM + Gender*URM + Series + HomeDept + c_first_appt + year + Degree +
  I(c_first_appt^2) + (1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
680 temp3 = lme(log(adj.salary) ~ Rank + Gender + URM + Gender*URM + Series + HomeDept + c_first_appt + year + Degree + I(
  c_first_appt^2), random = ~ 1|EMPLOYEE_ID, data = na.omit(newdat2))
681 fit3.13 = lmer(log(adj.salary) ~ Gender + URM + Gender*URM + Series + HomeDept + c_first_appt + year + Degree + I(
  c_first_appt^2) + (1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
682 fit3.23 = lmer(log(adj.salary) ~ Rank + URM + Series + HomeDept + c_first_appt + year + Degree + I(c_first_appt^2) +
  (1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
683 fit3.33 = lmer(log(adj.salary) ~ Rank + Gender + Series + HomeDept + c_first_appt + year + Degree + I(c_first_appt^2)
  + (1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
684 fit3.43 = lmer(log(adj.salary) ~ Rank + Gender + URM + Gender*URM + HomeDept + c_first_appt + year + Degree + I(
  c_first_appt^2) + (1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
685 fit3.53 = lmer(log(adj.salary) ~ Rank + Gender + URM + Gender*URM + Series + c_first_appt + year + Degree + I(
  c_first_appt^2) + (1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
686 fit3.63 = lmer(log(adj.salary) ~ Rank + Gender + URM + Gender*URM + Series + HomeDept + year + Degree + I(c_first_appt
  ^2) + (1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
687 fit3.73 = lmer(log(adj.salary) ~ Rank + Gender + URM + Gender*URM + Series + HomeDept + c_first_appt + Degree + I(
  c_first_appt^2) + (1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)

```

```

688 fit3.83 = lmer(log(adj.salary) ~ Rank + Gender + URM + Gender*URM + Series + HomeDept + c_first_appt + year + I(
      c_first_appt^2) + (1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
689 fit3.93 = lmer(log(adj.salary) ~ Rank + Gender + URM + Gender*URM + Series + HomeDept + c_first_appt + year + Degree +
      (1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
690 fit3.103 = lmer(log(adj.salary) ~ Rank + Gender + URM + Series + HomeDept + c_first_appt + year + Degree + I(
      c_first_appt^2) + (1|EMPLOYEE_ID), data = na.omit(newdat2), REML = F)
691 like.test = NULL
692 like.test[2] = anova(fit3.03, fit3.13)$Pr[2]
693 like.test[5] = anova(fit3.03, fit3.23)$Pr[2]
694 like.test[7] = anova(fit3.03, fit3.33)$Pr[2]
695 like.test[9] = anova(fit3.03, fit3.43)$Pr[2]
696 like.test[14] = anova(fit3.03, fit3.53)$Pr[2]
697 like.test[34] = anova(fit3.03, fit3.63)$Pr[2]
698 like.test[35] = anova(fit3.03, fit3.73)$Pr[2]
699 like.test[36] = anova(fit3.03, fit3.83)$Pr[2]
700 like.test[45] = anova(fit3.03, fit3.93)$Pr[2]
701 like.test[46] = anova(fit3.03, fit3.103)$Pr[2]
702 like.test[47] = NA
703 sum.tab3 = cbind(summary(fit3.03)$coefficients,
704                  pvalFormat(2 * pnorm(abs(summary(fit3.03)$coefficients[, 3]), lower.tail = FALSE)))
705 sum.tab3 = rbind(sum.tab3[1,], rep(NA, 4), sum.tab3[2:3,], rep(NA, 4), sum.tab3[4,], rep(NA, 4), sum.tab3[5, ], rep(NA
      , 4), sum.tab3[6:9,], rep(NA, 4), sum.tab3[10:30,], rep(NA, 4), sum.tab3[31:39,], rep(NA, 4), sum.tab3[40,])
706 sum.tab3 = cbind(sum.tab3, like.test)
707 sum.tab3 = matrix(sum.tab3, ncol = 5)
708 rownames(sum.tab3) = c("(Intercept)", "\\bfseries Rank", "~~~~Assoc Prof", "~~~~Professor", "\\bfseries Gender", "~~~~
      Female", "\\bfseries URM", "~~~~URM", "\\bfseries Series", "~~~~ADJ", "~~~~CLIN", "~~~~CLINx", "~~~~IR", "\\
      bfseries Home Department", "~~~~BIOENGINEERING", "~~~~CELL \\& MOL MED PROG", "~~~~CHEMISTRY \\& BIOCHEMISTRY",
      "~~~~DEP. OF REPRODUCTIVE MEDICINE", "~~~~DIVISION OF BIOLOGICAL SCI.", "~~~~FAMILY \\& PREVENTIVE MEDICINE",
      "~~~~MEDICINE", "~~~~MEDSCH/EMERG MED SVC", "~~~~NEUROSCIENCES", "~~~~OPHTHALMOLOGY", "~~~~ORTHOPAEDIC SURGERY",
      "~~~~PATHOLOGY", "~~~~PEDIATRICS", "~~~~PHARMACOLOGY", "~~~~PSYCHIATRY", "~~~~RADIATION MED \\& APPLIED SCI",
      "~~~~RADIOLOGY", "~~~~SCH OF PHARMACY AND PHARM. SCI", "~~~~SURGERY", "\\bfseries Years of experience at UCSD",
      "\\bfseries Fiscal Year", "\\bfseries Degree", "~~~~DO/PhD", "~~~~DPM", "~~~~MD", "~~~~MD/PhD", "~~~~NonDoc",
      "~~~~OtherDoc", "~~~~PharmD", "~~~~PhD", "\\bfseries Years of experience at UCSD$^{2}$", "\\bfseries Gender*URM",
      "~~~~Female*URM")
709 colnames(sum.tab3) = c("Coefficient", "Std Error", "Z-Value", "Z-Test", "LRT")
710 sum.tab3[, c(1, 2)] = paste('$', formatC(as.numeric(sum.tab3[, c(1, 2)]), digits = 4, format = 'f'), '$', sep = '')
711 sum.tab3[, c(3)] = paste('$', formatC(as.numeric(sum.tab3[, c(3)]), digits = 2, format = 'f'), '$', sep = '')
712 sum.tab3[, c(5)] = pvalFormat(sum.tab3[, c(5)])
713 sum.tab3[c(1, 3:4, 6, 8, 10:13, 15:33, 37:44, 47), 5] = rep('', 37)
714 sum.tab3[c(2, 5, 7, 9, 14, 36, 46), 1:4] = rep('', 28)
715 pdf(paste('residual3', '.pdf', sep = ''), height = 5, width = 6)
716 plot(fitted(fit3.03), scale(residuals(fit3.03)), col="dodgerblue", xlab = "Fitted Values", ylab = "Standardized
      Conditional Residuals")
717 abline(h=0)
718 dev.off()
719
720 sum2 = sum.tab3[c(6, 8, 47), 1:4]
721 rownames(sum2) = c("Gender(Female)", "URM(URM)", "Female*URM")
722
723 ###R square
724
725 Fmat1 <- model.matrix(eval(temp1$call$fixed)[-2], temp1$data)
726 VarF1 <- var(as.vector(fixef(temp1) %>% t(Fmat1)))
727 VarComp1 <- VarCorr(temp1)
728 VarRand1 <- as.numeric(VarComp1[rownames(VarComp1) != "Residual" & rownames(VarComp1) == "(Intercept)", "Variance"])
729 VarResid1 <- as.numeric(VarComp1[rownames(VarComp1) == "Residual", "Variance"])
730 Rm1 <- VarF1 / (VarF1 + sum(VarRand1) + VarResid1)
731 RXU1 <- 1 - as.numeric(VarCorr(temp1)[2, 1]) / as.numeric(VarCorr(temp0)[2, 1])
732 Rm1 = formatC(Rm1, digits = 3, format = 'f')
733 RXU1 = formatC(RXU1, digits = 3, format = 'f')
734
735 Fmat2 <- model.matrix(eval(temp2$call$fixed)[-2], temp2$data)
736 VarF2 <- var(as.vector(fixef(temp2) %>% t(Fmat2)))
737 VarComp2 <- VarCorr(temp2)
738 VarRand2 <- as.numeric(VarComp2[rownames(VarComp2) != "Residual" & rownames(VarComp2) == "(Intercept)", "Variance"])
739 VarResid2 <- as.numeric(VarComp2[rownames(VarComp2) == "Residual", "Variance"])
740 Rm2 <- VarF2 / (VarF2 + sum(VarRand2) + VarResid2)
741 RXU2 <- 1 - as.numeric(VarCorr(temp2)[2, 1]) / as.numeric(VarCorr(temp0)[2, 1])
742 Rm2 = formatC(Rm2, digits = 3, format = 'f')
743 RXU2 = formatC(RXU2, digits = 3, format = 'f')
744
745 Fmat3 <- model.matrix(eval(temp3$call$fixed)[-2], temp3$data)
746 VarF3 <- var(as.vector(fixef(temp3) %>% t(Fmat3)))
747 VarComp3 <- VarCorr(temp3)
748 VarRand3 <- as.numeric(VarComp3[rownames(VarComp3) != "Residual" & rownames(VarComp3) == "(Intercept)", "Variance"])
749 VarResid3 <- as.numeric(VarComp3[rownames(VarComp3) == "Residual", "Variance"])
750 Rm3 <- VarF3 / (VarF3 + sum(VarRand3) + VarResid3)
751 RXU3 <- 1 - as.numeric(VarCorr(temp3)[2, 1]) / as.numeric(VarCorr(temp0)[2, 1])
752 Rm3 = formatC(Rm3, digits = 3, format = 'f')

```

```

753 RXU3 = formatC(RXU3, digits = 3, format = 'f')
754
755 ###ranking table
756 year = c(2010, 2011, 2012, 2013, 2014)
757 rt.dep = function(i){
758   dat0 = newdat[which(newdat$year == year[i]),]
759   a10 = aggregate(dat0$adj.salary, list(dat0$HomeDept), median)
760   a2 = rank(-a10$x)
761   a10$x = prettyNum(formatC(a10$x, digits = 0, format = 'f'), big.mark = ",")
762   a10 = cbind(a10$x, a2)
763   colnames(a10) = c("Median", "Rank")
764   return(a10)
765 }
766 for (i in 1:5){
767   assign(paste('a', i, sep = ''), rt.dep(i))
768 }
769 aa = cbind(a1, a2, a3, a4, a5)
770 rownames(aa) = c("ANESTHESIOLOGY", "BIOENGINEERING", "CELL \\& MOL MED PROG", "CHEMISTRY \\& BIOCHEMISTRY", "DEP. OF
  REPRODUCTIVE MEDICINE", "DIVISION OF BIOLOGICAL SCI.", "FAMILY \\& PREVENTIVE MEDICINE", "MEDICINE", "MEDSCH/
  EMERG MED SVC", "NEUROSCIENCES", "OPHTHALMOLOGY", "ORTHOPAEDIC SURGERY", "PATHOLOGY", "PEDIATRICS", "
  PHARMACOLOGY", "PSYCHIATRY", "RADIATION MED \\& APPLIED SCI", "RADIOLOGY", "SCH OF PHARMACY AND PHARM. SCI", "
  SURGERY")
771 rank = as.numeric(aa[,2])
772 aa = aa[order(rank),]
773
774 rt.se = function(i){
775   dat0 = newdat[which(newdat$year == year[i]),]
776   b10 = aggregate(dat0$adj.salary, list(dat0$Series), median)
777   b2 = rank(-b10$x)
778   b10$x = prettyNum(formatC(b10$x, digits = 0, format = 'f'), big.mark = ",")
779   b10 = cbind(b10$x, b2)
780   colnames(b10) = c("Median", "Rank")
781   return(b10)
782 }
783 for (i in 1:5){
784   assign(paste('b', i, sep = ''), rt.se(i))
785 }
786 bb = cbind(b1, b2, b3, b4, b5)
787 rownames(bb) = c("FTE", "ADJ", "CLIN", "CLINx", "IR")
788 rank = as.numeric(bb[,2])
789 bb = bb[order(rank),]
790
791
792 rt.dg = function(i){
793   dat0 = newdat[which(newdat$year == year[i]),]
794   c10 = aggregate(dat0$adj.salary, list(dat0$Degree), median)
795   c2 = rank(-c10$x)
796   c10$x = prettyNum(formatC(c10$x, digits = 0, format = 'f'), big.mark = ",")
797   c10 = cbind(c10$x, c2)
798   colnames(c10) = c("Median", "Rank")
799   return(c10)
800 }
801 for (i in 1:5){
802   assign(paste('c', i, sep = ''), rt.dg(i))
803 }
804 c1 = rbind(c1[1:5,], rep("", 2), rep("", 2), c1[6:7,])
805 c2 = rbind(c2[1:5,], rep("", 2), c2[6:8,])
806 c3 = rbind(c3[1,], rep("", 2), c3[2:4,], rep("", 2), c3[5:7,])
807 c4 = rbind(c4[1,], rep("", 2), c4[2:4,], rep("", 2), c4[5:7,])
808 c5 = rbind(c5[1,], rep("", 2), rep("", 2), c5[2:7,])
809 c = cbind(c1, c2, c3, c4, c5)
810 rownames(c) = c("DO", "DO/PhD", "DPM", "MD", "MD/PhD", "NonDoc", "OtherDoc", "PharmD", "PhD")
811 rank = as.numeric(c[,2])
812 c = c[order(rank),]
813
814
815
816 rt.rk = function(i){
817   dat0 = newdat[which(newdat$year == year[i]),]
818   f10 = aggregate(dat0$adj.salary, list(dat0$Rank), median)
819   f2 = rank(-f10$x)
820   f10$x = prettyNum(formatC(f10$x, digits = 0, format = 'f'), big.mark = ",")
821   f10 = cbind(f10$x, f2)
822   colnames(f10) = c("Median", "Rank")
823   return(f10)
824 }
825 for (i in 1:5){
826   assign(paste('f', i, sep = ''), rt.rk(i))
827 }

```

```

828 f = cbind(f1, f2, f3, f4, f5)
829 rownames(f) = c("Asst Prof", "Assoc Prof", "Professor")
830 rank = as.numeric(f[,2])
831 f = f[order(rank),]
832
833 ###outliers
834 newdat2 = newdat[, c(1:5, 24, 28, 34, 46:47, 49)]
835 res.con = HLMresid(object = fit3.03, level = 1, type = "EB")
836 res.mar = HLMresid(object = fit3.03, level = "marginal")
837 sign.con = sign(res.con)
838 sign.mar = sign(res.mar)
839 h = mean(res.mar) + 2*sd(res.mar)
840 l = mean(res.mar) - 2*sd(res.mar)
841 fit = log(na.omit(newdat2)$adj.salary) - res.mar
842 gap = NULL
843 for (i in 1:nrow(na.omit(newdat2))) {
844   if (sign.mar[i] == 1) {
845     gap[i] = exp(fit[i] + h) - na.omit(newdat2)$adj.salary[i]
846   } else {
847     gap[i] = exp(fit[i] + l) - na.omit(newdat2)$adj.salary[i]
848   }
849 }
850 new = cbind(na.omit(newdat2), abs(res.con), abs(res.mar), sign.con, sign.mar, gap, fit)
851 names(new)[12:13] = c("res.con", "res.mar")
852 new = new[which(new$year == 2014),]
853 new = new[order(-new$res.mar),]
854 new$res.mar = new$res.mar * new$sign.mar
855 problem = NULL
856 for (i in 1:nrow(new)) {
857   if (new$sign.mar[i] == 1) {problem[i] = "High"}
858   else {problem[i] = "Low"}
859 }
860 new = cbind(new, problem)
861 later = new$fit
862 new$adj.salary = prettyNum(formatC(new$adj.salary, digits = 0, format = 'f'), big.mark = ",")
863 new$gap = prettyNum(formatC(new$gap, digits = 0, format = 'f'), big.mark = ",")
864 new$fit = prettyNum(formatC(exp(new$fit), digits = 0, format = 'f'), big.mark = ",")
865 new$res.mar = formatC(new$res.mar, digits = 3, format = 'f')
866 new$HomeDept = as.character(new$HomeDept)
867 for (i in 1:nrow(new)) {
868   if (new$HomeDept[i] == "FAMILY & PREVENTIVE MEDICINE") {
869     new$HomeDept[i] = "FAMILY \\& PREVENTIVE MEDICINE"
870   }
871 }
872 adj.new = new[which(as.numeric(new$res.mar) >= h | as.numeric(new$res.mar) <= l), c(2, 3, 4:8, 13, 11, 17, 16, 18)]
873 adj.new3 = new[which(new$res.mar < h & new$res.mar > l), c(2, 3, 4:8, 13, 11, 17, 16, 18)]
874 adj.new3$problem = "Median"
875 adj.new1 = adj.new[which(adj.new$problem == "High"),]
876 adj.new2 = adj.new[which(adj.new$problem == "Low"),]
877 adj.new1 = adj.new1[, -12]
878 adj.new2 = adj.new2[, -12]
879 adj.new1 = adj.new1[, -10]
880 adj.new2 = adj.new2[, -10]
881 names(adj.new1) = c("ID", "Degree", "Rank", "Series", "HomeDept", "Gender", "URM", "Residual", "Total Salary", "Gap")
882 adj.new2$HomeDept[c(10, 22, 25)] = rep("FAM \\& PREV MEDICINE", 3)
883 adj.new1$HomeDept[c(1, 50)] = rep("FAM \\& PREV MEDICINE", 2)
884 adj.new1$HomeDept[c(36, 37)] = rep("DEP. OF REPRO MEDICINE", 2)
885 rownames(adj.new1) <- NULL
886 names(adj.new2) = c("ID", "Degree", "Rank", "Series", "HomeDept", "Gender", "URM", "Residual", "Total Salary", "Gap")
887 rownames(adj.new2) <- NULL
888 residual = as.numeric(adj.new[, 8])
889
890 adj.new4 = new[which(new$res.mar < 0), c(2, 3, 4:8, 13, 11, 17, 16, 18)]
891 adj.new4 = adj.new4[1:100,]
892 adj.new4 = adj.new4[, -c(10, 12)]
893 names(adj.new4) = c("ID", "Degree", "Rank", "Series", "HomeDept", "Gender", "URM", "Residual", "Total Salary", "Gap")
894 rownames(adj.new4) <- NULL
895 adj.new4[26:100, 10] = rep(0, 75)
896 adj.new4$HomeDept[c(10, 22, 25, 26, 81)] = rep("FAM \\& PREV MEDICINE", 5)
897 adj.new4$HomeDept[c(44, 65, 67, 99)] = rep("DEP. OF REPRO MEDICINE", 4)
898 adj.new4$HomeDept[77] = "SCH OF PHARM \\& PHARM. SCI"
899
900 ###contingency table for 100
901 group = c(rep(0, 50), rep(1, 50))
902 t1 = cbind(adj.new4, group)
903 comb = NULL
904 for (i in 1:nrow(t1)) {
905   if (t1$Gender[i] == 'M' & t1$URM[i] == 'Not URM') {comb[i] = 'mn'}
906   else if (t1$Gender[i] == 'M' & t1$URM[i] == 'URM') {comb[i] = 'mu'}

```



```

907 else if (t1$Gender[i] == 'F' & t1$URM[i] == 'Not URM'){comb[i] = 'fn'}
908 else if (t1$Gender[i] == 'F' & t1$URM[i] == 'URM'){comb[i] = 'fu'}
909 }
910 t1 = cbind(t1, comb)
911 t1$group = factor(t1$group)
912 t1$comb = factor(t1$comb)
913 tt1 = demoTab(group ~ Degree + Rank + Gender + URM + comb + Series, DATA = t1, digits = 2)
914 ad = c("$0\\; (0.0\\%)$", "$0\\; (0.0\\%)$", "$0\\; (0.0\\%)$")
915 tt1 = rbind(tt1[1:22,], ad, tt1[23:30,])
916 rownames(tt1)[23] = '~~~~fu'
917 t2 = na.omit(newdat2)
918 t2 = t2[which(t2$year == 2014),]
919 t2 = cbind(t2, c(rep(0, 1000), rep(1, 337)))
920 names(t2)[12] = "group"
921 comb2 = NULL
922 for (i in 1:nrow(t2)){
923   if (t2$Gender[i] == 'M' & t2$URM[i] == 'Not URM'){comb2[i] = 'mn'}
924   else if (t2$Gender[i] == 'M' & t2$URM[i] == 'URM'){comb2[i] = 'mu'}
925   else if (t2$Gender[i] == 'F' & t2$URM[i] == 'Not URM'){comb2[i] = 'fn'}
926   else if (t2$Gender[i] == 'F' & t2$URM[i] == 'URM'){comb2[i] = 'fu'}
927 }
928 t2 = cbind(t2, comb2)
929 names(t2)[13] = "comb"
930 t2$comb = factor(t2$comb)
931 t2$group = factor(t2$group)
932 tt2 = demoTab(group ~ Degree + Rank + Gender + URM + comb + Series, DATA = t2, digits = 2)
933 ttt = cbind(tt1[, 3], tt2[, 3])
934 rownames(ttt)[21:25] = c("Gender * URM", "~~~~Female + Not URM", "~~~~Female + URM", "~~~~Male + Not URM", "~~~~Male +
URM")
935 colnames(ttt) = c("100 faculty members with the lowest residuals", "Whole dataset in 2013–2014")
936 ttt = ttt[-c(3, 4),]
937
938
939
940
941 ###contingency table for outliers
942 newnew = rbind(adj.new, adj.new3)
943 newnew$problem = factor(newnew$problem)
944 #data = na.omit(newdat2)
945 outTab = demoTab2(problem ~ Degree + Rank + Gender + URM + Series, DATA = newnew, digits = 2)[, -5]
946 outTab = cbind(outTab[, 2], outTab[, 3], outTab[, 1], outTab[, 4])
947 outTab = outTab[-c(3:4), ]
948 colnames(outTab)[1:4] = c('Below the 2.5th \\% ($n=25$)', '2.5 to 97.5 \\% ($n=1257$)', 'Above the 97.5th \\% ($n=55$)',
', 'Total')
949
950 uf = c(paste('$', table(adj.new2$Gender, adj.new2$URM)[2,2], '\\; (', formatC((table(adj.new2$Gender, adj.new2$URM)
[2,2]/(table(adj.new1$Gender, adj.new1$URM)[2,2]+table(adj.new3$Gender, adj.new3$URM)[2,2] + table(adj.
new2$Gender, adj.new2$URM)[2,2])) * 100, digits = 1, format = 'f'), '\\%', ')$'), sep = ''), paste('$', table(adj.
new3$Gender, adj.new3$URM)[2,2], '\\; (', formatC((table(adj.new3$Gender, adj.new3$URM)[2,2]/(table(adj.
new1$Gender, adj.new1$URM)[2,2]+table(adj.new3$Gender, adj.new3$URM)[2,2]+ table(adj.new2$Gender, adj.new2$URM)
[2,2])) * 100, digits = 1, format = 'f'), '\\%', ')$'), sep = ''), paste('$', table(adj.new1$Gender, adj.new1$URM)
[2,2], '\\; (', formatC((table(adj.new1$Gender, adj.new1$URM)[2,2]/(table(adj.new1$Gender, adj.new1$URM)[2,2]+
table(adj.new3$Gender, adj.new3$URM)[2,2]+ table(adj.new2$Gender, adj.new2$URM)[2,2])) * 100, digits = 1, format
= 'f'), '\\%', ')$'), sep = ''), paste('$', (table(adj.new1$Gender, adj.new1$URM)[2,2]+table(adj.new3$Gender, adj.
new3$URM)[2,2] + table(adj.new2$Gender, adj.new2$URM)[2,2]), '\\; ', sep = ''))
951
952 um = c(paste('$', table(adj.new2$Gender, adj.new2$URM)[1,2], '\\; (', formatC((table(adj.new2$Gender, adj.new2$URM)
[1,2]/(table(adj.new1$Gender, adj.new1$URM)[1,2]+table(adj.new3$Gender, adj.new3$URM)[1,2] + table(adj.
new2$Gender, adj.new2$URM)[1,2])) * 100, digits = 1, format = 'f'), '\\%', ')$'), sep = ''), paste('$', table(adj.
new3$Gender, adj.new3$URM)[1,2], '\\; (', formatC((table(adj.new3$Gender, adj.new3$URM)[1,2]/(table(adj.
new1$Gender, adj.new1$URM)[1,2]+table(adj.new3$Gender, adj.new3$URM)[1,2]+ table(adj.new2$Gender, adj.new2$URM)
[1,2])) * 100, digits = 1, format = 'f'), '\\%', ')$'), sep = ''), paste('$', table(adj.new1$Gender, adj.new1$URM)
[1,2], '\\; (', formatC((table(adj.new1$Gender, adj.new1$URM)[1,2]/(table(adj.new1$Gender, adj.new1$URM)[1,2]+
table(adj.new3$Gender, adj.new3$URM)[1,2]+ table(adj.new2$Gender, adj.new2$URM)[1,2])) * 100, digits = 1, format
= 'f'), '\\%', ')$'), sep = ''), paste('$', (table(adj.new1$Gender, adj.new1$URM)[1,2]+table(adj.new3$Gender, adj.
new3$URM)[1,2] + table(adj.new2$Gender, adj.new2$URM)[1,2]), '\\; ', sep = ''))
953
954 nf = c(paste('$', table(adj.new2$Gender, adj.new2$URM)[2,1], '\\; (', formatC((table(adj.new2$Gender, adj.new2$URM)
[2,1]/(table(adj.new1$Gender, adj.new1$URM)[2,1]+table(adj.new3$Gender, adj.new3$URM)[2,1] + table(adj.
new2$Gender, adj.new2$URM)[2,1])) * 100, digits = 1, format = 'f'), '\\%', ')$'), sep = ''), paste('$', table(adj.
new3$Gender, adj.new3$URM)[2,1], '\\; (', formatC((table(adj.new3$Gender, adj.new3$URM)[2,1]/(table(adj.
new1$Gender, adj.new1$URM)[2,1]+table(adj.new3$Gender, adj.new3$URM)[2,1]+ table(adj.new2$Gender, adj.new2$URM)
[2,1])) * 100, digits = 1, format = 'f'), '\\%', ')$'), sep = ''), paste('$', table(adj.new1$Gender, adj.new1$URM)
[2,1], '\\; (', formatC((table(adj.new1$Gender, adj.new1$URM)[2,1]/(table(adj.new1$Gender, adj.new1$URM)[2,1]+
table(adj.new3$Gender, adj.new3$URM)[2,1]+ table(adj.new2$Gender, adj.new2$URM)[2,1])) * 100, digits = 1, format
= 'f'), '\\%', ')$'), sep = ''), paste('$', (table(adj.new1$Gender, adj.new1$URM)[2,1]+table(adj.new3$Gender, adj.
new3$URM)[2,1] + table(adj.new2$Gender, adj.new2$URM)[2,1]), '\\; ', sep = ''))
955
956 nm = c(paste('$', table(adj.new2$Gender, adj.new2$URM)[1,1], '\\; (', formatC((table(adj.new2$Gender, adj.new2$URM)

```

```

[1,1]/(table(adj.new1$Gender, adj.new1$URM)[1,1]+table(adj.new3$Gender, adj.new3$URM)[1,1] + table(adj.
new2$Gender, adj.new2$URM)[1,1])) * 100, digits = 1, format = 'f', '\\%', '$', sep = ''), paste('$', table(adj.
new3$Gender, adj.new3$URM)[1,1], '\\; (' ,formatC((table(adj.new3$Gender, adj.new3$URM)[1,1]/(table(adj.
new1$Gender, adj.new1$URM)[1,1]+table(adj.new3$Gender, adj.new3$URM)[1,1]+ table(adj.new2$Gender, adj.new2$URM)
[1,1])) * 100, digits = 1, format = 'f', '\\%', '$', sep = ''), paste('$', table(adj.new1$Gender, adj.new1$URM)
[1,1], '\\; (' ,formatC((table(adj.new1$Gender, adj.new1$URM)[1,1]/(table(adj.new1$Gender, adj.new1$URM)[1,1]+
table(adj.new3$Gender, adj.new3$URM)[1,1]+ table(adj.new2$Gender, adj.new2$URM)[1,1])) * 100, digits = 1, format
= 'f', '\\%', '$', sep = ''), paste('$', (table(adj.new1$Gender, adj.new1$URM)[1,1]+table(adj.new3$Gender, adj.
new3$URM)[1,1] + table(adj.new2$Gender, adj.new2$URM)[1,1]), '\\; ', sep = ''))
957
958 title = c("", "", "", "")
959 addln = rbind(title, uf, um, nf, nm)
960
961 outTab = rbind(outTab[1:18, ], addln, outTab[19:24, ])
962 rownames(outTab)[c(14:15, 19:23)] = c("~~~~Male", "~~~~Female", "URM * Gender", "~~~~URM + Female", "~~~~URM + Male",
"~~~~Not URM + Female", "~~~~Not URM + Male")
963
964 pdf(paste('residual4', '.pdf', sep = ''), height = 5, width = 6)
965 plot(later, scale(as.numeric(new$res.mar)), col="dodgerblue", xlab = "Fitted Values", ylab = "Standardized Marginal
Residuals", main = " Residual Plot in fiscal year 2013–2014")
966 abline(h=0)
967 dev.off()
968
969 ###contingency table before the model
970 ntable = matrix(NA, nrow = 8, ncol = 4)
971 a = as.numeric(quantile(newdat[which(newdat$year == 2014,)]$adj.salary, c(0.05, 0.95))[1])
972 b = as.numeric(quantile(newdat[which(newdat$year == 2014,)]$adj.salary, c(0.05, 0.95))[2])
973 t11 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary <= a & newdat$URM == "URM"),])
974 t21 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary <= a & newdat$URM == "Not URM"),])
975 t12 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > a & newdat$URM == "URM" & newdat$adj.salary <= b),])
976 t22 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > a & newdat$URM == "Not URM" & newdat$adj.salary <= b
),])
977 t13 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > b & newdat$URM == "URM"),])
978 t23 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > b & newdat$URM == "Not URM"),])
979
980 t31 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary <= a & newdat$Gender == "F"),])
981 t41 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary <= a & newdat$Gender == "M"),])
982 t32 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > a & newdat$Gender == "F" & newdat$adj.salary <= b)
,])
983 t42 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > a & newdat$Gender == "M" & newdat$adj.salary <= b)
,])
984 t33 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > b & newdat$Gender == "F"),])
985 t43 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > b & newdat$Gender == "M"),])
986
987 t51 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary <= a & newdat$URM == "URM" & newdat$Gender == "F"),])
988 t61 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary <= a & newdat$URM == "URM" & newdat$Gender == "M"),])
989 t71 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary <= a & newdat$URM == "Not URM" & newdat$Gender == "F"
),])
990 t81 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary <= a & newdat$URM == "Not URM" & newdat$Gender == "M"
),])
991
992 t51 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary <= a & newdat$URM == "URM" & newdat$Gender == "F"),])
993 t61 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary <= a & newdat$URM == "URM" & newdat$Gender == "M"),])
994 t71 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary <= a & newdat$URM == "Not URM" & newdat$Gender == "F"
),])
995 t81 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary <= a & newdat$URM == "Not URM" & newdat$Gender == "M"
),])
996
997 t52 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > a & newdat$URM == "URM" & newdat$Gender == "F" &
newdat$adj.salary <= b),])
998 t62 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > a & newdat$URM == "URM" & newdat$Gender == "M" &
newdat$adj.salary <= b),])
999 t72 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > a & newdat$URM == "Not URM" & newdat$Gender == "F" &
newdat$adj.salary <= b),])
1000 t82 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > a & newdat$URM == "Not URM" & newdat$Gender == "M" &
newdat$adj.salary <= b),])
1001
1002 t53 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > b & newdat$URM == "URM" & newdat$Gender == "F"),])
1003 t63 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > b & newdat$URM == "URM" & newdat$Gender == "M"),])
1004 t73 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > b & newdat$URM == "Not URM" & newdat$Gender == "F"
),])
1005 t83 = nrow(newdat[which(newdat$year == 2014 & newdat$adj.salary > b & newdat$URM == "Not URM" & newdat$Gender == "M"
),])
1006
1007 t14 = nrow(newdat[which(newdat$year == 2014 & newdat$URM == "URM"),])
1008 t24 = nrow(newdat[which(newdat$year == 2014 & newdat$URM == "Not URM"),])
1009 t34 = nrow(newdat[which(newdat$year == 2014 & newdat$Gender == "F"),])
1010 t44 = nrow(newdat[which(newdat$year == 2014 & newdat$Gender == "M"),])
1011 t54 = nrow(newdat[which(newdat$year == 2014 & newdat$URM == "URM" & newdat$Gender == "F"),])

```

```

1012 t64 = nrow(newdat[which(newdat$year == 2014 & newdat$URM == "URM" & newdat$Gender == "M"),])
1013 t74 = nrow(newdat[which(newdat$year == 2014 & newdat$URM == "Not URM" & newdat$Gender == "F"),])
1014 t84 = nrow(newdat[which(newdat$year == 2014 & newdat$URM == "Not URM" & newdat$Gender == "M"),])
1015
1016
1017 ntable[, 1] = c(paste(t11, " (", formatC(t11/t14*100, digits = 0, format = 'f'), "%", sep = ''), paste(t21, "
(", formatC(t21/t24*100, digits = 0, format = 'f'), "%", sep = ''), paste(t31, " (", formatC(t31/t34*100,
digits = 0, format = 'f'), "%", sep = ''), paste(t41, " (", formatC(t41/t44*100, digits = 0, format = 'f')
, "%", sep = ''), paste(t51, " (", formatC(t51/t54*100, digits = 0, format = 'f'), "%", sep = ''),
paste(t61, " (", formatC(t61/t64*100, digits = 0, format = 'f'), "%", sep = ''), paste(t71, " (", formatC
(t71/t74*100, digits = 0, format = 'f'), "%", sep = ''), paste(t81, " (", formatC(t81/t84*100, digits = 0,
format = 'f'), "%", sep = ''))
1018
1019 ntable[, 2] = c(paste(t12, " (", formatC(t12/t14*100, digits = 0, format = 'f'), "%", sep = ''), paste(t22, "
(", formatC(t22/t24*100, digits = 0, format = 'f'), "%", sep = ''), paste(t32, " (", formatC(t32/t34*100,
digits = 0, format = 'f'), "%", sep = ''), paste(t42, " (", formatC(t42/t44*100, digits = 0, format = 'f')
, "%", sep = ''), paste(t52, " (", formatC(t52/t54*100, digits = 0, format = 'f'), "%", sep = ''),
paste(t62, " (", formatC(t62/t64*100, digits = 0, format = 'f'), "%", sep = ''), paste(t72, " (", formatC
(t72/t74*100, digits = 0, format = 'f'), "%", sep = ''), paste(t82, " (", formatC(t82/t84*100, digits = 0,
format = 'f'), "%", sep = ''))
1020
1021 ntable[, 3] = c(paste(t13, " (", formatC(t13/t14*100, digits = 0, format = 'f'), "%", sep = ''), paste(t23, "
(", formatC(t23/t24*100, digits = 0, format = 'f'), "%", sep = ''), paste(t33, " (", formatC(t33/t34*100,
digits = 0, format = 'f'), "%", sep = ''), paste(t43, " (", formatC(t43/t44*100, digits = 0, format = 'f')
, "%", sep = ''), paste(t53, " (", formatC(t53/t54*100, digits = 0, format = 'f'), "%", sep = ''),
paste(t63, " (", formatC(t63/t64*100, digits = 0, format = 'f'), "%", sep = ''), paste(t73, " (", formatC
(t73/t74*100, digits = 0, format = 'f'), "%", sep = ''), paste(t83, " (", formatC(t83/t84*100, digits = 0,
format = 'f'), "%", sep = ''))
1022
1023 ntable[, 4] = c(t14, t24, t34, t44, t54, t64, t74, t84)
1024 rownames(ntable) = c("URM", "Not URM", "Female", "Male", "URM + Female", "URM + Male", "Not URM + Female", "Not URM +
Male")
1025 colnames(ntable) = c("Below the 5th %% ($\\leq{\\$98,126}$)", "5 to 95 %% ($\\$98,126$, $\\$534,921$)", "Above the
95th %% ($>{\\$534,921}$)", "Total")

```