# Analysis

Andrew

## Information for Finding Model

$\alpha = 0.05$

Model with interaction:

$Y_{ijk} = \mu_{..} + \gamma_i + \delta_j + (\gamma\delta)_{ij} + \epsilon_{ijk}$

Model with factor A and factor B:

$Y_{ijk} = \mu_{..} + \gamma_i + \delta_j + \epsilon_{ijk}$

Model with factor A only:

$Y_{ijk} = \mu_{..} + \gamma_i + \epsilon_{ijk}$

Model with factor B only:

$Y_{ijk} = \mu_{..} + \delta_j + \epsilon_{ijk}$

Final model after conducting F-statistic tests:

$Y_{ijk} = \mu_{..} + \gamma_i + \delta_j + \epsilon_{ijk}$

With constraints: $\sum \gamma_i = 0$, $\sum \delta_j = 0$

$H_0$ : Use full model

$H_a$ : Use reduced model

## Model Selection Table

| Model | Interaction Effect | Factor A Effect (Region) | Factor B Effect (House Size) |
|---|---|---|---|
| $R^2$ | 0.002130082 | 0.005056614 | 0.007065998 |
| $F_s$ | 0.9463521 | 6.7747240 | 3.1620011 |
| $p$ | 0.41742959 | 0.00934836 | 0.02380460 |
| $p \leq \alpha$ | FALSE | TRUE | TRUE |
| Include | FALSE | TRUE | TRUE |

## Analysis - Pairwise CIs for Factor A

| | x |
|---|---|
| Bonferroni | 2.642 |
| Tukey | 2.854 |

| | x |
|---|---|
| Scheffe | 3.332 |

We can pick Bonferroni, Tukey, or Scheffe multiplier. We pick the smallest multiplier, Bonferroni (2.642), for more precision.

| | Mean | Lower | Upper | Significant |
|---|---|---|---|---|
| $\mu_{1.} - \mu_{6.}$ | 3673.709 | -6580.687 | 13928.105 | FALSE |
| $\mu_{2.} - \mu_{1.}$ | 358.6957 | -1866.5624 | 2583.9538 | FALSE |
| $\mu_{5.} - \mu_{2.}$ | 1145.627 | -5538.021 | 7829.275 | FALSE |
| $\mu_{3.} - \mu_{5.}$ | 1268.732 | -5498.189 | 8035.653 | FALSE |
| $\mu_{4.} - \mu_{3.}$ | 189.2347 | -3103.4618 | 3481.9311 | FALSE |
| $\mu_{4.} - \mu_{6.}$ | 6635.998 | -3848.402 | 17120.398 | FALSE |

## Analysis - Pairwise CIs for Factor B

| | x |
|---|---|
| Bonferroni | 2.501 |
| Tukey | 2.572 |
| Scheffe | 2.799 |

We can pick Bonferroni, Tukey, or Scheffe multiplier. We pick the smallest multiplier, Bonferroni (2.501), for more precision.

| | Mean | Lower | Upper | Significant |
|---|---|---|---|---|
| $\mu_{.2} - \mu_{.4}$ | -81.73664 | -6362.72614 | 6199.25286 | FALSE |
| $\mu_{.1} - \mu_{.2}$ | 717.9974 | -5970.7391 | 7406.7339 | FALSE |
| $\mu_{.3} - \mu_{.1}$ | 1552.632 | -3400.924 | 6506.188 | FALSE |
| $\mu_{.3} - \mu_{.4}$ | 2188.893 | -2198.516 | 6576.301 | FALSE |

## Appendix

```
# Setup
alpha <- 0.05

data <- read.csv("insurance.csv")
selected_columns <- c("charges", "children", "region")
data <- subset(data, select = selected_columns)
colnames(data) <- c("Y", "A", "B")

AB <- lm(Y ~ A * B, data = data)
A.B <- lm(Y ~ A + B, data = data)
A <- lm(Y ~ A, data = data)
B <- lm(Y ~ B, data = data)
N <- lm(Y ~ 1, data = data)
```

```r
# Functions
# Give me means
find.means = function(data,fun.name = mean){
  a = length(unique(data[,2]))
  b = length(unique(data[,3]))
  means.A = by(data[,1], data[,2], fun.name)
  means.B = by(data[,1],data[,3],fun.name)
  means.AB = by(data[,1],list(data[,2],data[,3]),fun.name)
  MAB = matrix(means.AB,nrow = b, ncol = a, byrow = TRUE)
  colnames(MAB) = names(means.A)
  rownames(MAB) = names(means.B)
  MA = as.numeric(means.A)
  names(MA) = names(means.A)
  MB = as.numeric(means.B)
  names(MB) = names(means.B)
  MAB = t(MAB)
  results = list(A = MA, B = MB, AB = MAB)
  return(results)
}

# Partial R^2
Partial.R2 = function(small.model,big.model){
  SSE1 = sum(small.model$residuals^2)
  SSE2 = sum(big.model$residuals^2)
  PR2 = (SSE1 - SSE2) / SSE1
  return(PR2)
}

# Give me multipliers
find.mult = function(alpha,a,b,dfSSE,g,group){
  if(group == "A"){
    Tuk = round(qtukey(1-alpha,a,dfSSE)/sqrt(2),3)
    Bon = round(qt(1-alpha/(2*g), dfSSE ) ,3)
    Sch = round(sqrt((a-1)*qf(1-alpha, a-1, dfSSE)),3)
  }else if(group == "B"){
    Tuk = round(qtukey(1-alpha,b,dfSSE)/sqrt(2),3)
    Bon = round(qt(1-alpha/(2*g), dfSSE ) ,3)
    Sch = round(sqrt((b-1)*qf(1-alpha, b-1, dfSSE)),3)
  }else if(group == "AB"){
    Tuk = round(qtukey(1-alpha,a*b,dfSSE)/sqrt(2),3)
    Bon = round(qt(1-alpha/(2*g), dfSSE ) ,3)
    Sch = round(sqrt((a*b-1)*qf(1-alpha, a*b-1, dfSSE)),3)
  }
  results = c(Bon, Tuk,Sch)
  names(results) = c("Bonferroni","Tukey","Scheffe")
  return(results)
}

# Give me CI
give.me.CI = function(data,MSE,equal.weights = TRUE,multiplier,group,cs){
   if(sum(cs) != 0 & sum(cs !=0 ) != 1){
    return("Error - you did not input a valid contrast")
  }else{
```

```r
  the.means = find.means(data)
  the.ns =find.means(data,length)
  nt = nrow(data)
  a = length(unique(data[,2]))
  b = length(unique(data[,3]))
  if(group =="A"){
    if(equal.weights == TRUE){
      a.means = rowMeans(the.means$AB)
      est = sum(a.means*cs)
      mul = rowSums(1/the.ns$AB)
      SE = sqrt(MSE/b^2 * (sum(cs^2*mul)))
      N = names(a.means)[cs!=0]
      CS = paste("(",cs[cs!=0],")",sep = "")
      fancy = paste(paste(CS,N,sep =""),collapse = "+")
      names(est) = fancy
    } else{
      a.means = the.means$A
      est = sum(a.means*cs)
      SE = sqrt(MSE*sum(cs^2*(1/the.ns$A)))
      N = names(a.means)[cs!=0]
      CS = paste("(",cs[cs!=0],")",sep = "")
      fancy = paste(paste(CS,N,sep =""),collapse = "+")
      names(est) = fancy
    }
  }else if(group == "B"){
    if(equal.weights == TRUE){
      b.means = colMeans(the.means$AB)
      est = sum(b.means*cs)
      mul = colSums(1/the.ns$AB)
      SE = sqrt(MSE/a^2 * (sum(cs^2*mul)))
      N = names(b.means)[cs!=0]
      CS = paste("(",cs[cs!=0],")",sep = "")
      fancy = paste(paste(CS,N,sep =""),collapse = "+")
      names(est) = fancy
    } else{
      b.means = the.means$B
      est = sum(b.means*cs)
      SE = sqrt(MSE*sum(cs^2*(1/the.ns$B)))
      N = names(b.means)[cs!=0]
      CS = paste("(",cs[cs!=0],")",sep = "")
      fancy = paste(paste(CS,N,sep =""),collapse = "+")
      names(est) = fancy
    }
  } else if(group == "AB"){
    est = sum(cs*the.means$AB)
    SE = sqrt(MSE*sum(cs^2/the.ns$AB))
    names(est) = "someAB"
  }
  the.CI = est + c(-1,1)*multiplier*SE
  results = c(est,the.CI)
  names(results) = c(names(est),"lower bound","upper bound")
  return(results)
}
```

```r
}
```

```r
# Model Selection Table
model_selection <- data.frame(
  "Model" = c(
    "Interaction Effect",
    "Factor A Effect (Region)",
    "Factor B Effect (House Size)"
  )
)

model_selection$"$R^2$" <- c(
  Partial.R2(A.B, AB),
  Partial.R2(B, A.B),
  Partial.R2(A, A.B)
)

model_selection$"$F_s$" <- c(
  anova(A.B, AB)$F[2],
  anova(B, A.B)$F[2],
  anova(A, A.B)$F[2]
)

model_selection$"$p$" <- c(
  anova(A.B, AB)$"Pr(>F)"[2],
  anova(B, A.B)$"Pr(>F)"[2],
  anova(A, A.B)$"Pr(>F)"[2]
)

model_selection$"$p \\leq \\alpha$" <- model_selection$"$p$" <= alpha

model_selection$Include <- model_selection$"$p \\leq \\alpha$"

knitr::kable(t(model_selection))
```

```r
# Get SSE
all.models = list(AB,A.B,A,B,N)
SSE = t(as.matrix(sapply(all.models,function(M) sum(M$residuals^2))))
colnames(SSE) = c("AB","(A+B)","A","B","Empty/Null")
rownames(SSE) = "SSE"
```

```r
# Analysis Setup
# Get relavent values for CI based on model choice
n_T <- nrow(data)
a <- length(unique(data$A))
b <- length(unique(data$B))
sse <- SSE["SSE", "(A+B)"]
df_sse <- n_T - a - b + 1
mse <- sse / df_sse
```

```r
# Get Multiplier for Factor A
all.mult <- find.mult(alpha = alpha, a = a, b = b, dfSSE = df_sse, g = 6,
                      group = "A")
```

```r
the.mult <- min(all.mult)
knitr::kable(all.mult)
```

```r
# Do Confidence Intervals for Factor A
# Get combinations of all potential confidence intervals
CIs.A <- data.frame(Values = c("Mean", "Lower", "Upper"))
CIs.A$"$\\mu_{1.} - \\mu_{6.}$" <- give.me.CI(
  data, mse, equal.weights = TRUE,
  the.mult, "A", c(1, 0, 0, 0, 0, -1)
)
CIs.A$"$\\mu_{2.} - \\mu_{1.}$" <- give.me.CI(
  data, mse, equal.weights = TRUE,
  the.mult, "A", c(-1, 1, 0, 0, 0, 0)
)

CIs.A$"$\\mu_{5.} - \\mu_{2.}$" <- give.me.CI(
  data, mse, equal.weights = TRUE,
  the.mult, "A", c(0, -1, 0, 0, 1, 0)
)

CIs.A$"$\\mu_{3.} - \\mu_{5.}$" <- give.me.CI(
  data, mse, equal.weights = TRUE,
  the.mult, "A", c(0, 0, 1, 0, -1, 0)
)

CIs.A$"$\\mu_{4.} - \\mu_{3.}$" <- give.me.CI(
  data, mse, equal.weights = TRUE,
  the.mult, "A", c(0, 0, -1, 1, 0, 0)
)

CIs.A$"$\\mu_{4.} - \\mu_{6.}$" <- give.me.CI(
  data, mse, equal.weights = TRUE,
  the.mult, "A", c(0, 0, 0, 1, 0, -1)
)

CIs.A <- as.data.frame(t(CIs.A))
colnames(CIs.A) <- CIs.A[1, ]
CIs.A <- CIs.A[-1, ]

CIs.A$Significant <- FALSE

knitr::kable(CIs.A)
```

```r
# Get Multiplier for Factor B
all.mult <- find.mult(alpha = alpha, a = a, b = b, dfSSE = df_sse, g = 4,
                      group = "B")
the.mult <- min(all.mult)
knitr::kable(all.mult)
```

```r
# Do Confidence Intervals for Factor B
# Get combinations of all potential confidence intervals
CIs.B <- data.frame(Values = c("Mean", "Lower", "Upper"))
```

```r
CIs.B$"$\\mu_{.2} - \\mu_{.4}$" <- give.me.CI(
  data, mse, equal.weights = TRUE,
  the.mult, "B", c(0, 1, 0, -1)
)

CIs.B$"$\\mu_{.1} - \\mu_{.2}$" <- give.me.CI(
  data, mse, equal.weights = TRUE,
  the.mult, "B", c(1, -1, 0, 0)
)

CIs.B$"$\\mu_{.3} - \\mu_{.1}$" <- give.me.CI(
  data, mse, equal.weights = TRUE,
  the.mult, "B", c(-1, 0, 1, 0)
)

CIs.B$"$\\mu_{.3} - \\mu_{.4}$" <- give.me.CI(
  data, mse, equal.weights = TRUE,
  the.mult, "B", c(0, 0, 1, -1)
)

CIs.B <- as.data.frame(t(CIs.B))
colnames(CIs.B) <- CIs.B[1, ]
CIs.B <- CIs.B[-1, ]

CIs.B$Significant <- FALSE

knitr::kable(CIs.B)
```