

## Paul Kogan-Midterm2

Paul Kogan

2021/10/19

```
pacman::p_load(leaps, MASS)
rjct <- \(lvl) paste("<= alpha =", lvl,
  "so reject the null\n\t\t\tthe hypothesis that the")
conc <- \(test, hyp, lvl = .05, cat = T) {
  val <- try(test$p.value)
  lvl <- ifelse(length(lvl) == 0, .05, lvl)
  val <- ifelse(length(val) == 0, test[, 5], val)
  str <- paste("conclusion: p-value =", format(round(val, 4), nsmall = 4),
    ifelse(val <= lvl, rjct(lvl), rjct(lvl) %>%
      str_replace_all(c("<=" = ">", "so" = "so fail to"))), hyp)
  if (cat) cat(str)
  invisible(str)
}
```

**1**

```
data <- read.table("d_logret_6stocks.txt", header = T)
attach(data)

(fit1 <- lm(Pfizer ~ AmerExp))
cat("\na\t Intercept:\t", fit1$coefficients[[1]],
    "\n \t AmerExp:\t\t", fit1$coefficients[[2]], "\nb\n\t")
(table <- anova(fit1))
conc(table, "effects are not significant")
test <- cor.test(Pfizer, AmerExp)
cat("\nc\n\tCorrelation:\t", cor(Pfizer, AmerExp),
    "\n\tCor Test:")
test
conc(test, "correlation is zero")
cat("\nd")

cat("\ne")
Group1 <- Exxon
Group2 <- AmerExp
Group3 <- Pfizer
y <- c(Group1, Group2, Group3)
x <- c(rep(1, 64), rep(2, 64), rep(3, 64))
fit2 <- lm(y ~ factor(x))
(test2 <- anova(fit2))
conc(test2, "means are the same")
```

##

```

## Call:
## lm(formula = Pfizer ~ AmerExp)
##
## Coefficients:
## (Intercept)      AmerExp
##   -0.004156      0.162127
##
##
## a)      Intercept: -0.004155557
##        AmerExp:      0.1621268
## b)
## Analysis of Variance Table
##
## Response: Pfizer
##           Df    Sum Sq    Mean Sq F value  Pr(>F)
## AmerExp    1 0.001492 0.00149172   2.8838 0.09449 .
## Residuals 62 0.032071 0.00051728
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## conclusion: p-value = 0.0945 > alpha = 0.05 so fail to reject the null
##              hypothesis that the effects are not significant
## c)
## Correlation:      0.2108206
## Cor Test:
## Pearson's product-moment correlation
##
## data:  Pfizer and AmerExp
## t = 1.6982, df = 62, p-value = 0.09449
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.03690097  0.43413240
## sample estimates:
##           cor
## 0.2108206
##
## conclusion: p-value = 0.0945 > alpha = 0.05 so fail to reject the null
##              hypothesis that the correlation is zero
## d)
## e)Analysis of Variance Table
##
## Response: y
##           Df    Sum Sq    Mean Sq F value Pr(>F)
## factor(x)   2 0.001914 0.00095695   1.4657 0.2335
## Residuals 189 0.123396 0.00065289
## conclusion: p-value = 0.2335 > alpha = 0.05 so fail to reject the null
##              hypothesis that the means are the same

```

2

```

library(MASS)
model <- lm(y ~ 0 + x1 + x2 + x3 + x4, data = cement)
best <- coef(leaps::regsubsets(y ~ ., data = cement, nvmax = 2), 1:2)[[2]]
cat("\na) Coefficients:\t", model$coefficients,
    "\nb) Adj. R Squared:\t", summary(model)$adj.r.squared,

```

```
"\nc) Best:\ty = ", best[1], " + ", best[2], "x1 + ", best[3], "x2")
```

```
##
```

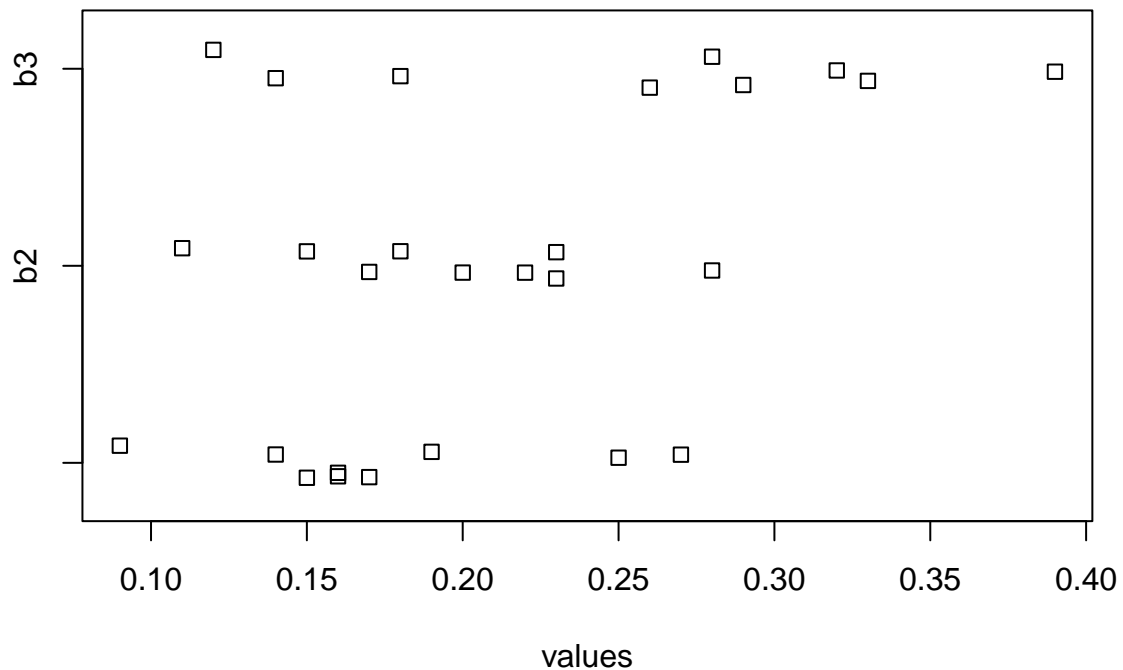
```
## a) Coefficients: 2.193046 1.153326 0.7585091 0.4863193
```

```
## b) Adj. R Squared: 0.9993724
```

```
## c) Best: y = 52.57735 + 1.468306 x1 + 0.6622505 x2
```

3

```
tredecula <- c(0.09, 0.17, 0.19, 0.16, 0.27, 0.15, 0.25, 0.16, 0.14)
tredecassini <- c(0.28, 0.15, 0.22, 0.11, 0.20, 0.17, 0.23, 0.18, 0.23)
tredecim <- c(0.28, 0.29, 0.14, 0.18, 0.32, 0.39, 0.33, 0.26, 0.12)
t_tredecula <- shapiro.test(tredecula)
t_tredecassini <- shapiro.test(tredecassini)
t_tredecim <- shapiro.test(tredecim)
list <- list(tredecula, b2 = tredecassini, b3 = tredecim)
stack <- stack(list)
bart <- bartlett.test(list)
cat("\na")
t_tredecula
conc(t_tredecula, "tredecula weights are normal", 0.1)
t_tredecassini
conc(t_tredecassini, "tredecassini weights are normal", 0.1)
t_tredecim
conc(t_tredecim, "tredecim weights are normal", 0.1)
cat("\nb")
bart
conc(bart, "variances are the same", 0.1)
cat("\nc")
stripchart(values ~ ind, data = stack, method = "jitter")
```



```
cat("\nd")
(test <- anova(lm(values ~ ind, data = stack)))
conc(test, "the weights are the same")
```

```
##
## a)
## Shapiro-Wilk normality test
##
## data:  tredecula
## W = 0.92755, p-value = 0.4584
##
## conclusion: p-value = 0.4584 > alpha = 0.1 so fail to reject the null
##           hypothesis that the tredecula weights are normal
## Shapiro-Wilk normality test
##
## data:  tredecassini
## W = 0.9844, p-value = 0.983
##
## conclusion: p-value = 0.9830 > alpha = 0.1 so fail to reject the null
##           hypothesis that the tredecassini weights are normal
## Shapiro-Wilk normality test
##
## data:  tredecim
## W = 0.94143, p-value = 0.597
##
```

```
## conclusion: p-value = 0.5970 > alpha = 0.1 so fail to reject the null
##          hypothesis that the tredecim weights are normal
## b)
## Bartlett test of homogeneity of variances
##
## data: list
## Bartlett's K-squared = 3.3285, df = 2, p-value = 0.1893
##
## conclusion: p-value = 0.1893 > alpha = 0.1 so fail to reject the null
##          hypothesis that the variances are the same
## c
## dAnalysis of Variance Table
##
## Response: values
##      Df Sum Sq Mean Sq F value Pr(>F)
## ind      2 0.031874 0.0159370  3.4205 0.04932 *
## Residuals 24 0.111822 0.0046593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## conclusion: p-value = 0.0493 <= alpha = 0.05 so reject the null
##          hypothesis that the the weights are the same
```

4

```
cat("\na")
(obs <- t(matrix(c(4, 6, 10, 18, 14, 20, 5, 10), nrow = 2)))
test <- chisq.test(obs)
```

```
## Warning in chisq.test(obs): Chi-squared approximation may be incorrect
```

```
pval <- test$p.value
cat("\nb")
test
conc(test, "effects are independent")
cat("\nc")
test$expected
```

```
##
## a)      [,1] [,2]
## [1,]     4     6
## [2,]    10    18
## [3,]    14    20
## [4,]     5    10
##
## b)
## Pearson's Chi-squared test
##
## data: obs
## X-squared = 0.36341, df = 3, p-value = 0.9477
##
## conclusion: p-value = 0.9477 > alpha = 0.05 so fail to reject the null
##          hypothesis that the effects are independent
```

```
## c)          [,1]      [,2]
## [1,]  3.793103  6.206897
## [2,] 10.620690 17.379310
## [3,] 12.896552 21.103448
## [4,]  5.689655  9.310345
```