Portfolio 3 - fMRI Regression

Nanna Bernth, Sebastian Scott, Roxana Petrache, Fredrik Sejr & Signe M. R. Holdgaard
2/22/2018

Load data and packages

```
#Load packages
library(tidyverse)
library(stats)
library(lmerTest)
library(ggplot2)
library(reshape)

#Set working directory
setwd("~/Dropbox/Uni/2 semester/Ekseperimental Methods 2/R-code/Portfolio 3")

#Load data
fmri<-as.matrix(read.csv("portfolio_assignment3_aud_fmri_data37.csv", header=FALSE
))

#making it a time-series
fmri2<-ts(fmri)

#Make it into a data frame
fmri_df <- data.frame(fmri2)

#load in design
fmrides<-as.matrix(read.csv("portfolio_assignment3_aud_fmri_design.csv", header=FA
LSE))

#making it into a time-series
fmrides2<-ts(fmrides)

#Make it into a data frame
fmri_des_df <- data.frame(fmrides2)

#Add time serie count to data frame
fmri_des_df <- mutate(fmri_des_df, "Time_serie" = 1:400)
```

Tasks

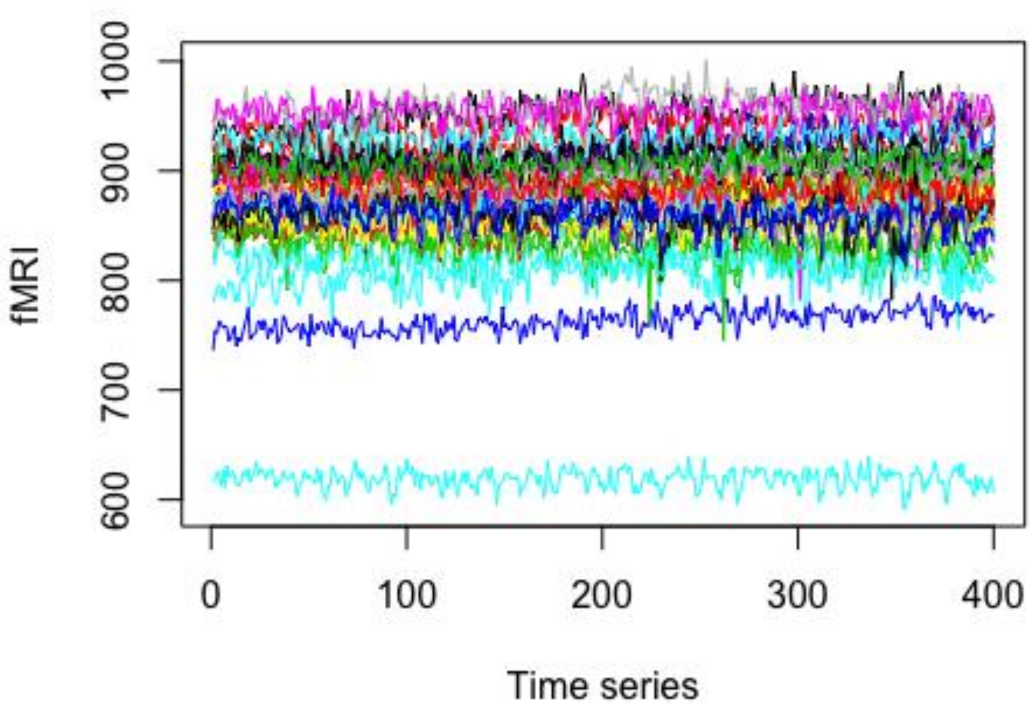
In this excercise we are going to look at data from an unpublished fMRI experiment. 400 whole-brain EPI images were acquired for each participant, but in this assignment, we will analyse a time-series from a single voxel in auditory cortex (transverse temporal gyrus, MNI coordinate: [-46,-20,6], with all time-points converted into a vector). A total of 37 participants were scanned. They listened to two types of stories (fiction and factual). A model of the hemodynamic response to the different story types are included in a separate file. The task is to perform a regresion with these two different independent variables using different models and also adding an additional covariate.

Initial figures

1. Make two figures:

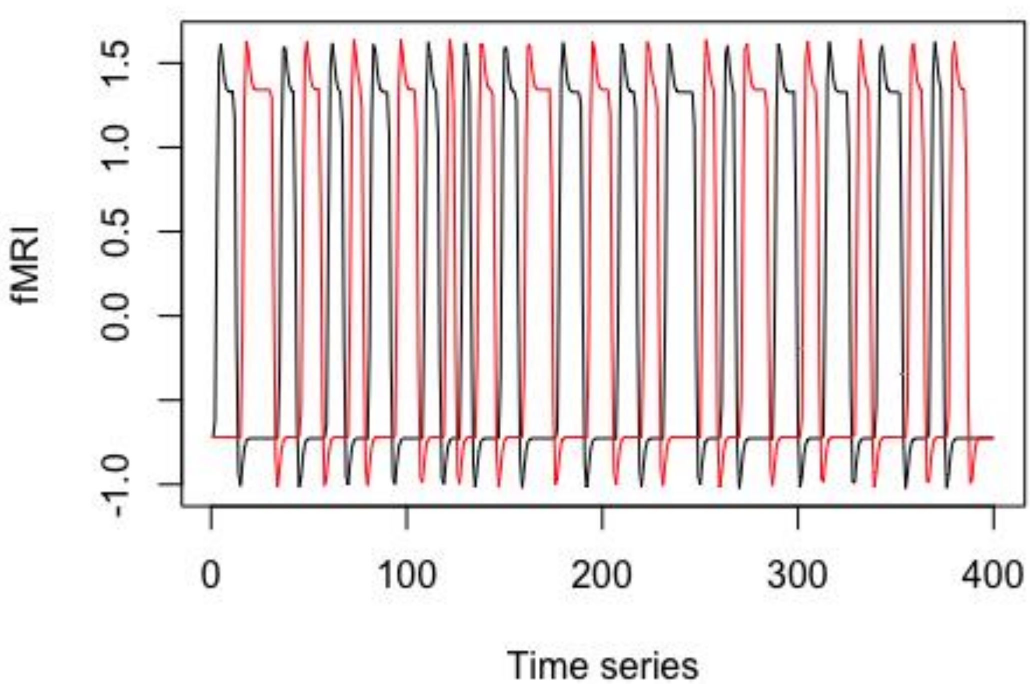
1.a. A figure with lineplots of the data from all participants as a function of time in one figure. Note how much the baseline signal can vary between participants.

```
#Make a plot of the data as a function of time
ts.plot(fmri2, xlab = "Time series", ylab = "fMRI", col = 1:37)
```



1.b. A figure lineplots with the model covariates.

```r
#Make a plot of the model as a function of time
ts.plot(fmrides2, xlab = "Time series", ylab = "fMRI", col = 1:2)
```



Investigating model

2. How many stories did the participants listen to in each condition?

```r
#Count the number og peaks from the plot above
```

2 response: V1 had 15 stories. V2 had 15 stories.

3.a. Are the two model covariates correlated?

```r
#Make a correlation test of the two covariates from the design
cor.test(fmrides2[,1], fmrides2[,2])

## 
##  Pearson's product-moment correlation
## 
## data:  fmrides2[, 1] and fmrides2[, 2]
```

```
## t = -12.894, df = 398, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6084775 -0.4697617
## sample estimates:
##        cor
## -0.5428111
```

3.a response: They have a correlation of -0.54, which means that they have a strong negative correlation.

3.b. Have the covariates been mean-centered?

```
#Use summary function to get the stats for the two columns
summary(fmrides2)

##        V1                    V2
##  Min.   :-1.0229000   Min.   :-1.016100
##  1st Qu.:-0.7265900   1st Qu.:-0.719063
##  Median :-0.7264300   Median :-0.718940
##  Mean   :-0.0000013   Mean   : 0.000002
##  3rd Qu.: 1.3276000   3rd Qu.: 1.341300
##  Max.   : 1.6240000   Max.   : 1.638500
```

3.b response: The two covariates' means are both very close to 0, which means that they are probably mean-centered.

4. Please report the percentage of shared variance in the two covariates.

```
#Calculate R^2 to find the shared variance
(-0.5428111)^2

## [1] 0.2946439

#Get it in percent
((-0.5428111)^2)*100

## [1] 29.46439
```

4 Response: They have a shared variance of 29.5%.

Analysis

Single participant

5. Pick one participant’s data set.

5. Response We pick participant 31

Conduct 6 analyses using lm(): #5.a. Fit the model as it is, including intercept.

```
#Take column 31 from the fmri2 dataframe and make it a new dataframe
data_31 <- data.frame(fmri2[,31])

#Change column names of the data frame
colnames(data_31) <- c("V31")

#Make a model with the participants data and the two covariates including an inter
cept
model_31 <- lm(V31 ~ fmri_des_df[,1] + fmri_des_df[,2], data = data_31)

#Summarise model
summary(model_31)

##
## Call:
## lm(formula = V31 ~ fmri_des_df[, 1] + fmri_des_df[, 2], data = data_31)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.497  -5.908  -0.230   5.582  33.889
##
## Coefficients:
##                  Estimate Std. Error  t value Pr(>|t|)
## (Intercept)      847.9225     0.4352 1948.371   <2e-16 ***
## fmri_des_df[, 1]   5.0567     0.5188    9.746   <2e-16 ***
## fmri_des_df[, 2]   5.7562     0.5188   11.095   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.704 on 397 degrees of freedom
## Multiple R-squared:  0.2645, Adjusted R-squared:  0.2608
## F-statistic: 71.38 on 2 and 397 DF,  p-value: < 2.2e-16
```

5.b. Fit the model as it is, excluding intercept.

```
#Model without intercept is done by adding minus 1 to the model
model_31_noincept <- lm(V31 ~ fmri_des_df[,1] + fmri_des_df[,2] - 1, data = data_3
1)

#Summarise model
summary(model_31_noincept)

##
## Call:
## lm(formula = V31 ~ fmri_des_df[, 1] + fmri_des_df[, 2] - 1, data = data_31)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  824.4  842.0  847.7  853.5  881.8
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## fmri_des_df[, 1]    5.056     50.673   0.100    0.921
## fmri_des_df[, 2]    5.758     50.673   0.114    0.910
##
## Residual standard error: 850.1 on 398 degrees of freedom
## Multiple R-squared:  3.761e-05,  Adjusted R-squared:  -0.004987
## F-statistic: 0.007485 on 2 and 398 DF,  p-value: 0.9925
```

5.c. Fit only the 1st covariate as a model.

```
#Make a model with only the first covariate added
model_cov1 <- lm(V31 ~ fmri_des_df[,1], data = data_31)

#Summarise model
summary(model_cov1)

##
## Call:
## lm(formula = V31 ~ fmri_des_df[, 1], data = data_31)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.9841  -6.4953   0.3919   6.1226  30.4829
##
## Coefficients:
##                  Estimate Std. Error  t value Pr(>|t|)
## (Intercept)      847.9225     0.4975 1704.410  < 2e-16 ***
## fmri_des_df[, 1]   1.9322     0.4981    3.879 0.000123 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 9.95 on 398 degrees of freedom
## Multiple R-squared:  0.03643,    Adjusted R-squared:  0.03401
## F-statistic: 15.05 on 1 and 398 DF,  p-value: 0.0001227
```

5.d. Fit only the 2nd covariate as a model.

```
#Make a model with only the second covariate
model_cov2 <- lm(V31 ~ fmri_des_df[,2], data = data_31)

#Summarise model
summary(model_cov2)

##
## Call:
## lm(formula = V31 ~ fmri_des_df[, 2], data = data_31)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.7685  -5.8304   0.0922   6.2807  30.2425
##
## Coefficients:
##                  Estimate Std. Error  t value Pr(>|t|)
## (Intercept)      847.9225     0.4839 1752.404  < 2e-16 ***
## fmri_des_df[, 2]   3.0113     0.4845    6.216 1.29e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.677 on 398 degrees of freedom
## Multiple R-squared:  0.08849,    Adjusted R-squared:  0.0862
## F-statistic: 38.64 on 1 and 398 DF,  p-value: 1.29e-09
```

The residuals represent the variance left when fitting a model. They are thus data that have been “cleaned” from the variance explained by the model. We can use those “cleaned” data to fit another model on. This is similar to using a type III sum of squares approach to your statistics.

5.e. Fit the 2nd covariate to the residuals from analysis 5.c., the 1st covariate only analysis

```
#Make a new column in the design data frame which contains the residuals from the
model with only the first covariate
fmri_des_df <- mutate(fmri_des_df, "Residuals_1cov" = residuals(model_cov1))
```

```
#Make a new model with the residuals from the model with only the first covariate
predicted by the second covariate
model_res_fit1 <- lm(Residuals_1cov ~ V2, data = fmri_des_df)

#Summarise model
summary(model_res_fit1)

##
## Call:
## lm(formula = Residuals_1cov ~ V2, data = fmri_des_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.0800  -5.7530  -0.0136   5.4401  30.4001
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.303e-06  4.541e-01   0.000        1
## V2           4.060e+00  4.546e-01   8.931   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.081 on 398 degrees of freedom
## Multiple R-squared:  0.1669, Adjusted R-squared:  0.1648
## F-statistic: 79.76 on 1 and 398 DF,  p-value: < 2.2e-16
```

5.f. Fit the 1st covariate to the resistuals from 5.d., the 2nd covariate only analysis

```
#Make a new column in the design data frame which contains the residuals from the
model with only the second covariate
fmri_des_df <- mutate(fmri_des_df, "Residuals_2cov" = residuals(model_cov2))

#Make a new model with the residuals from the model with only the second covariate
predicted by the first covariate
model_res_fit2 <- lm(Residuals_2cov ~ V1, data = fmri_des_df)

#Summarise model
summary(model_res_fit2)

##
## Call:
## lm(formula = Residuals_2cov ~ V1, data = fmri_des_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.1904  -5.9874  -0.1728   5.5317  30.8335
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.811e-06  4.497e-01   0.000        1
## V1          3.567e+00  4.503e-01   7.921 2.37e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.994 on 398 degrees of freedom
## Multiple R-squared:  0.1362, Adjusted R-squared:  0.134
## F-statistic: 62.75 on 1 and 398 DF,  p-value: 2.374e-14
```

5.g. Does the order in which the predictor variables are fitted to the data matter for the estimates? If it does, what can explain this?

5.g Response: Yes because they explain some of the same variance which we calculated in task 4, 29.5%. The shared variance is attributed to the first predictor. Which means that the first predictor added in the model will explain the shared variance of 29.5%.

Group level analyses

6. Fit the full model to each of the 37 participants' data and extract the coefficients for each participant. (hint: the full participant data frame can be set as outcome. Alternatively, you can change the data

structure and use lmList from assignement 1).

```
#Make a model with the data from all participants as outcome and predicted by the two covariates
full_model <- lm(fmri ~ fmri_des_df$V1 + fmri_des_df$V2, data = fmri_df)

#Extract the coefficients from the model
coef_full_model <- coefficients(full_model)

#Print the coefficients from the model
coef_full_model

##                        V1         V2         V3         V4         V5
## (Intercept)    867.327495 919.434995 831.229993 890.132497 618.772496
## fmri_des_df$V1   9.582330   5.773264   7.340939   2.682228   5.872990
## fmri_des_df$V2   8.926516   6.104578   8.233871   3.133700   6.041478
##                        V6         V7         V8         V9        V10
## (Intercept)    903.394998 880.842498 868.254998 952.377501 936.684994
```

```
## fmri_des_df$V1   3.209499   2.606846   3.828594   2.422324   6.400593
## fmri_des_df$V2   3.230518   2.499852   3.530697   0.992906   7.207420
##                       V11        V12        V13        V14        V15
## (Intercept)    847.879996 916.202498 806.272499 900.069996 884.099998
## fmri_des_df$V1   6.989003   3.464375   4.626842   3.304558   3.566603
## fmri_des_df$V2   6.538979   3.385156   3.595325   3.914555   3.495208
##                       V16        V17        V18        V19        V20
## (Intercept)    959.169996 907.019997 875.312497 879.467498 762.874998
## fmri_des_df$V1   5.566010   3.756718   5.436519   2.814987   2.829585
## fmri_des_df$V2   5.802205   3.873774   5.276158   2.626577   2.658036
##                       V21        V22        V23        V24        V25
## (Intercept)    926.434997 868.142496 860.584997 848.987497 910.402496
## fmri_des_df$V1   7.351865   5.963731   5.827429   3.880726   6.534162
## fmri_des_df$V2   6.394320   5.905760   5.165241   4.050404   6.399795
##                       V26        V27        V28        V29        V30
## (Intercept)    850.392497 829.929999 855.849995 863.177495 954.617495
## fmri_des_df$V1   4.849484   4.997005   7.818241   5.474936   6.648170
## fmri_des_df$V2   4.703407   3.951056   7.767283   5.903824   6.628171
##                       V31        V32        V33        V34        V35
## (Intercept)    847.922495 889.667497 859.482494 885.882499 905.452497
## fmri_des_df$V1   5.056695   4.891110   4.526764   4.904306   4.095500
## fmri_des_df$V2   5.756162   4.607988   6.001619   3.930084   4.176857
##                      V36        V37
## (Intercept)    857.69999 815.187497
## fmri_des_df$V1  10.52120   4.760363
## fmri_des_df$V2  10.63107   4.646695
```

6.a. Test the two individual hypotheses that the set of coefficient from each covariate is different from zero across the whole group (similar to assignment 1).

```
#Add coefficient from before to a matrix and transpose matrix
coef_matrix <- t(as.matrix(coef_full_model))

#Change column names
colnames(coef_matrix) <- c("Intercept", "coef_V1", "coef_V2")

#Add the matrix to a data frame
coef_full_df <- data.frame(coef_matrix)

#Make a t-test that tests if the first covariate is significantly different from z
ero
t.test(coef_full_df$coef_V1, mu=0)

##
##  One Sample t-test
```

```
##
## data:  coef_full_df$coef_V1
## t = 16.607, df = 36, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  4.512224 5.767586
## sample estimates:
## mean of x
##  5.139905
```

```r
#Make a t-test that tests if the second covariate is significantly different from
zero
t.test(coef_full_df$coef_V2, mu=0)
```

```
##
##  One Sample t-test
##
## data:  coef_full_df$coef_V2
## t = 15.603, df = 36, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  4.413274 5.731982
## sample estimates:
## mean of x
##  5.072628
```

6.a Response: The conducted t-tests show that the set of coefficients from each covariate are significantly different from zero p < .001 across the whole group.

Make a contrast that investigates the difference between the two covariates, i.e. the two types of stories (hint: subtraction).

```r
#Make a new column that substracts the two coeffecients from the covariates in ord
er to make the contrast
coef_full_df$contrast <- coef_full_df$coef_V1 - coef_full_df$coef_V2

#Print values
coef_full_df$contrast
```

```
##  [1]  0.65581370 -0.33131441 -0.89293270 -0.45147183 -0.16848855
##  [6] -0.02101973  0.10699381  0.29789701  1.42941758 -0.80682690
## [11]  0.45002325  0.07921988  1.03151654 -0.60999652  0.07139554
## [16] -0.23619571 -0.11705603  0.16036159  0.18841019  0.17154882
## [21]  0.95754501  0.05797112  0.66218773 -0.16967800  0.13436743
## [26]  0.14607779  1.04594858  0.05095722 -0.42888830  0.01999890
## [31] -0.69946688  0.28312193 -1.47485532  0.97422218 -0.08135714
## [36] -0.10987261  0.11366801
```

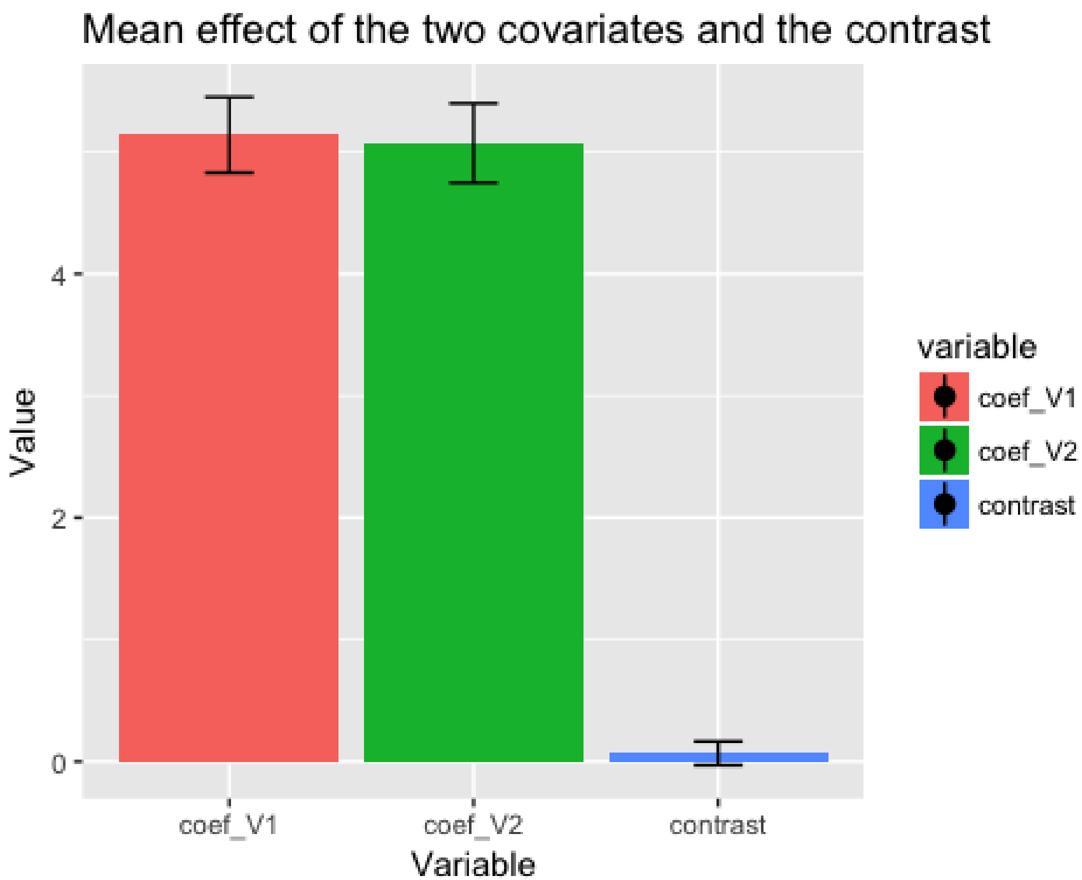
6.b. Test the hypothesis that the contrast is different from zero across participants.

```
#Make a t-test that tests if the contrast is significantly different from zero
t.test(coef_full_df$contrast, mu=0)

##
##  One Sample t-test
##
## data:  coef_full_df$contrast
## t = 0.69615, df = 36, p-value = 0.4908
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.1287197  0.2632734
## sample estimates:
##  mean of x
## 0.06727684
```

6.b response: The conducted t-test shows that the contrast is not significantly different from zero p = .49.

6.c. Make a bar diagram including the mean effect of the two coefficents and the contrast, including error bars (indicating standard error of mean).

```
#Make a new data frame containing column 2:4 from the coefficients data frame
coef_df_prep <- coef_full_df[,2:4]

#Change dataframe from wide format to long format
coef_long_df <- melt(coef_df_prep)

## Using  as id variables

#Make a bar plot of the mean effect of the two coefficents and the contrast
ggplot(coef_long_df, aes(x=variable, y=value, fill = variable))+
  geom_bar(stat = "summary", fun.y=mean)+
  stat_summary(fun.y=mean)+
  geom_errorbar(stat = "summary", fun.data = mean_se, width = 0.2)+ #add errorbars
with standard error
  labs(title = "Mean effect of the two covariates and the contrast", x = "Variable
", y = "Value") #Add titles

## Warning: Removed 3 rows containing missing values (geom_pointrange).
```



Adding a covariate

7.a. For each partipant, add a covariate that models the effect of time (hint: 1:400).

```
#Make a model with time aspect added as predictor so we get a coefficient with tim
e for each particiant
time_model <- lm(fmri ~ fmri_des_df$V1 + fmri_des_df$V2 + fmri_des_df$Time_serie,
data = fmri_df)

#Extract coefficients from the model above and add to a new data frame
coef_time_model <- coefficients(time_model)

#Print data frame
coef_time_model

##                                 V1           V2           V3
## (Intercept)            868.057757118 917.24249031 837.55250945
```

```
## fmri_des_df$V1           9.550016393   5.87028137   7.06117026
## fmri_des_df$V2           8.884649034   6.23027923   7.87138862
## fmri_des_df$Time_serie  -0.003642207   0.01093519  -0.03153375
##                                   V4           V5          V6           V7
## (Intercept)             876.79804983 6.185361e+02 900.4884695 8.807089e+02
## fmri_des_df$V1            3.27227124 5.883448e+00   3.3381112 2.612759e+00
## fmri_des_df$V2            3.89819103 6.055029e+00   3.3971556 2.507513e+00
## fmri_des_df$Time_serie    0.06650597 1.178802e-03   0.0144964 6.664460e-04
##                                   V8          V9          V10           V11
## (Intercept)             848.49767678 941.9348796 927.60082261  8.479025e+02
## fmri_des_df$V1            4.70284663   2.8844049   6.80256334  6.988008e+00
## fmri_des_df$V2            4.66342486   1.5916030   7.72823394  6.537690e+00
## fmri_des_df$Time_serie    0.09854025   0.0520829   0.04530758 -1.121534e-04
##                                 V12          V13          V14          V15
## (Intercept)             892.7402380 795.61412533 8.992625e+02 880.03295301
## fmri_des_df$V1            4.5025700   5.09847001 3.340289e+00   3.74656833
## fmri_des_df$V2            4.7302952   4.20639168 3.960850e+00   3.72837991
## fmri_des_df$Time_serie    0.1170187   0.05315897 4.027377e-03   0.02028451
##                                  V16          V17          V18          V19
## (Intercept)             954.46701161 912.16343362 8.741432e+02 868.40969542
## fmri_des_df$V1            5.77411465   3.52912280 5.488261e+00   3.30428970
## fmri_des_df$V2            6.07183717   3.57888981 5.343197e+00   3.26054326
## fmri_des_df$Time_serie    0.02345628  -0.02565305 5.831993e-03   0.05515113
##                                  V20          V21          V22          V23
## (Intercept)             751.24373875 9.257719e+02 871.82995391 8.601728e+02
## fmri_des_df$V1            3.34426289 7.381207e+00   5.80056280 5.845666e+00
## fmri_des_df$V2            3.32488007 6.432338e+00   5.69435050 5.188871e+00
## fmri_des_df$Time_serie    0.05801127 3.307345e-03  -0.01839131 2.055644e-03
##                                   V24          V25          V26
## (Intercept)             850.598664559 903.30991996 838.88551273
## fmri_des_df$V1            3.809432408   6.84800561   5.35866325
## fmri_des_df$V2            3.958032297   6.80642684   5.36312562
## fmri_des_df$Time_serie   -0.008035748   0.03537444   0.05739144
##                                  V27          V28           V29
## (Intercept)             835.16177946 859.96791931 863.316852239
## fmri_des_df$V1            4.76550105   7.63602432   5.468769527
## fmri_des_df$V2            3.65110767   7.53119433   5.895834694
## fmri_des_df$Time_serie   -0.02609367  -0.02053828  -0.000695047
##                                  V30          V31          V32          V33
## (Intercept)             952.53847881 8.464594e+02 873.19771742 861.80835251
## fmri_des_df$V1            6.74016583 5.121437e+00   5.61989054   4.42384595
## fmri_des_df$V2            6.74736567 5.840045e+00   5.55223457   5.86827321
## fmri_des_df$Time_serie    0.01036916 7.297286e-03   0.08214354  -0.01160029
##                                  V34          V35          V36          V37
## (Intercept)             892.35984168 900.53874714 864.30288350 821.65375618
## fmri_des_df$V1            4.61768694   4.31293097  10.22902039   4.47423387
## fmri_des_df$V2            3.55872490   4.45857226  10.25251065   4.27597100
## fmri_des_df$Time_serie   -0.03230595   0.02450748  -0.03293212  -0.03225067
```

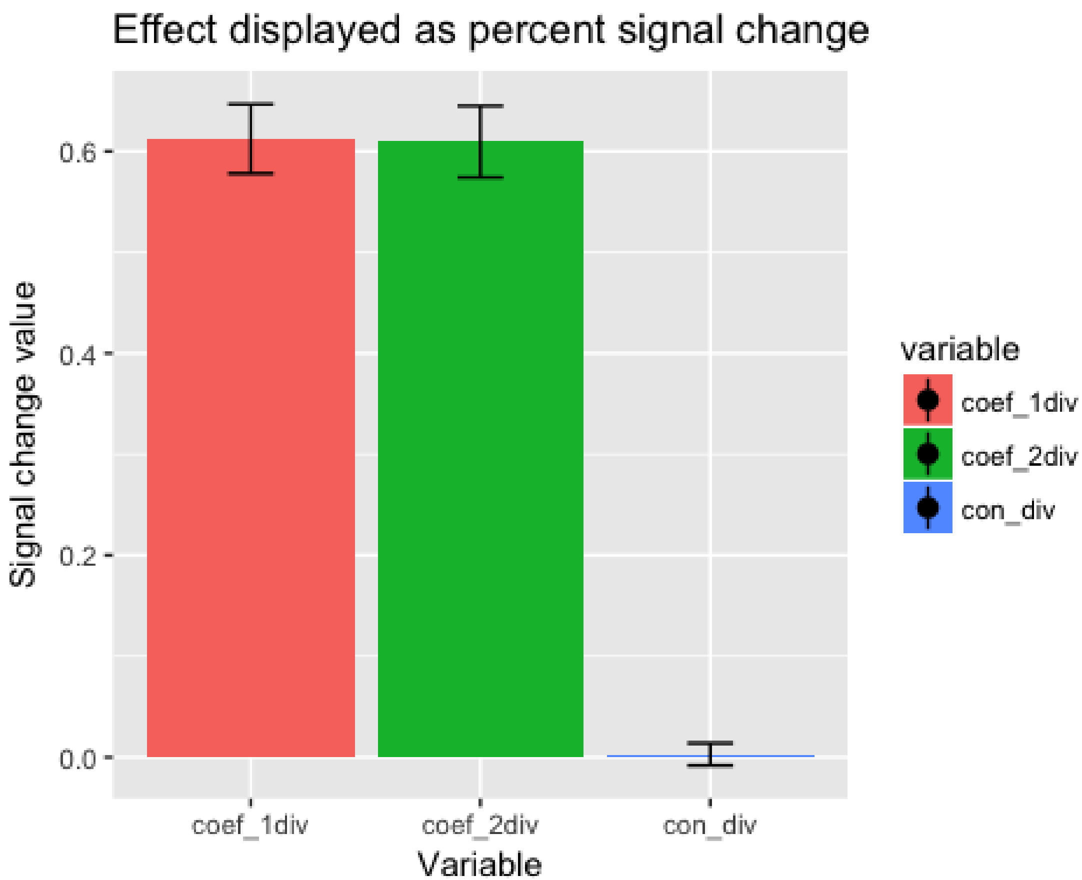
7.a. Does that improve the group results in term of higher t-values?

```
#Make model into a matrix and transpose matrix
coef_time_matrix <- t(as.matrix(coef_time_model))

#Change rownames of data frame
colnames(coef_time_matrix) <- c("Intercept", "coef_V1", "coef_V2", "Time")

#Make it into a data frame instead of matrix
coef_time_df <- data.frame(coef_time_matrix)

#Make a new t-test with the coefficients from the first covariate
t.test(coef_time_df$coef_V1, mu=0)

##
##  One Sample t-test
##
## data:  coef_time_df$coef_V1
## t = 18.3, df = 36, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  4.699321 5.870780
## sample estimates:
## mean of x
##  5.285051

#Make a new t-test with the coefficients from the second covariate
t.test(coef_time_df$coef_V2, mu=0)

##
##  One Sample t-test
##
## data:  coef_time_df$coef_V2
## t = 17.609, df = 36, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  4.654803 5.866570
## sample estimates:
## mean of x
##  5.260686
```

7.a Response: The group results does improve in terms of higher t-values. The first covariate t-value starts at 16.6 and changes to 18.3. The second covariate starts at 15.6 and changes to 17.6.

8. Make a bar diagram like the above, but display effects as percent signal change (hint: percent signal change is slope divided by intercept).

```
#Add contrast to data frame
coef_time_df$contrast <- coef_time_df$coef_V1-coef_time_df$coef_V2

#Make three new columns that calculates the percent signal change which is each sl
ope (the two covariates + the contrast) divided by the intercept.
coef_time_df <- mutate(coef_time_df, "coef_1div" = (coef_V1/Intercept)*100, "coef_
2div" = (coef_V2/Intercept)*100, "con_div" = (contrast/Intercept)*100)

#Extract column 6:8 from the data set (the columns calculated above)
coef_time_prep <- coef_time_df[,6:8]

#Change it from wide format to long format
coef_time_df_long <- melt(coef_time_prep)

## Using  as id variables

#Make a bar diagram with the percent signal change
ggplot(coef_time_df_long, aes(x=variable, y=value, fill = variable))+
  geom_bar(stat = "summary", fun.y=mean)+
  stat_summary(fun.y=mean)+
  geom_errorbar(stat = "summary", fun.data = mean_se, width = 0.2)+ #add errorbars
showing the standard error
  labs(title = "Effect displayed as percent signal change", x = "Variable", y = "S
ignal change value") #add titles

## Warning: Removed 3 rows containing missing values (geom_pointrange).
```


Effect displayed as percent signal change
Signal change value
0.6
0.4
0.2
0.0
coef_1div
coef_2div
con_div
Variable
variable
coef_1div
coef_2div
con_div