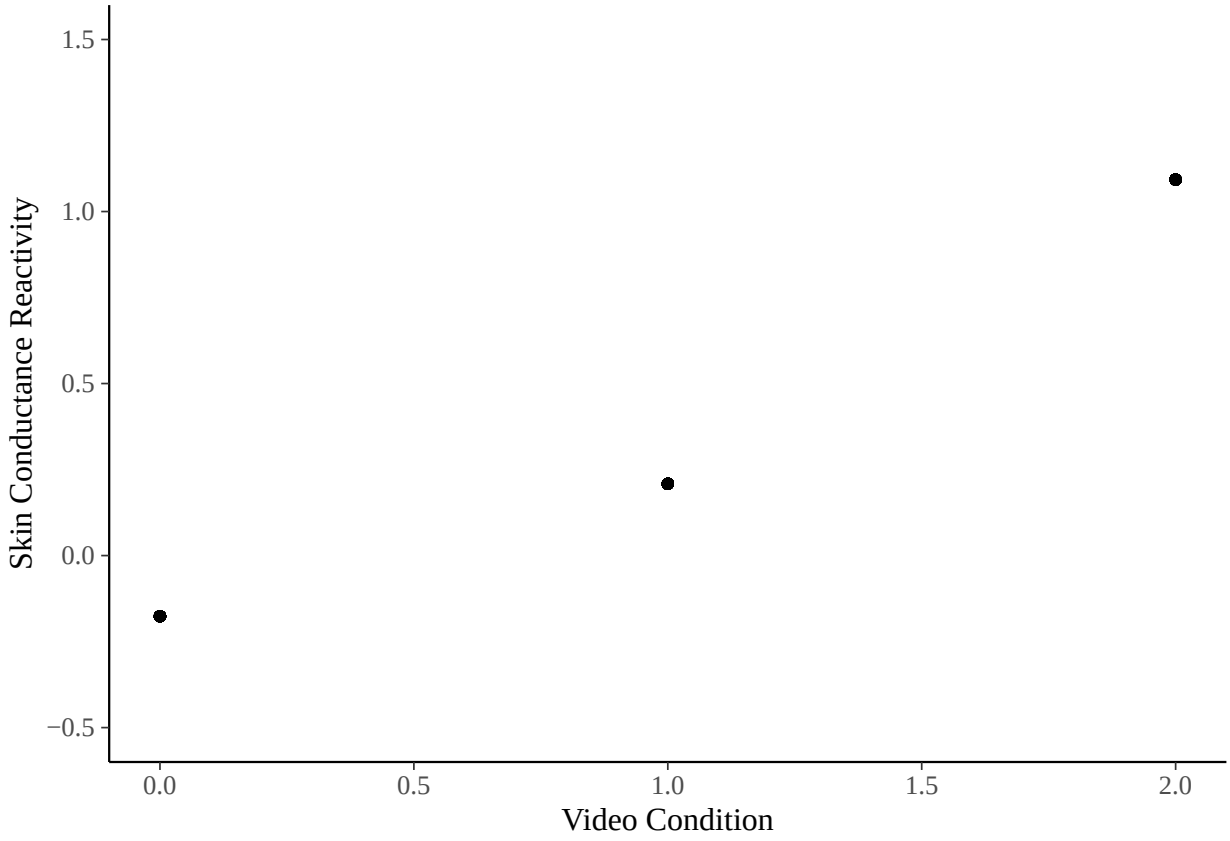
Midterm 1

Enxhi Buxheli

3/7/2019




```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000129 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.29 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
```

```
## Chain 1:
## Chain 1:  Elapsed Time: 0.099854 seconds (Warm-up)
## Chain 1:                0.085919 seconds (Sampling)
## Chain 1:                0.185773 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 3e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.3 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.081196 seconds (Warm-up)
## Chain 2:                0.091343 seconds (Sampling)
## Chain 2:                0.172539 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 1.6e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.16 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.079243 seconds (Warm-up)
## Chain 3:                0.083802 seconds (Sampling)
## Chain 3:                0.163045 seconds (Total)
```

```
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 2e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.086638 seconds (Warm-up)
## Chain 4:                0.094423 seconds (Sampling)
## Chain 4:                0.181061 seconds (Total)
## Chain 4:
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      SCDBradSelfReport1_mean ~ anxcond + storycond
##  observations: 129
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept)  0.0    0.1
## anxcond      0.5    0.2
## storycond   -0.1    0.2
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.8    0.1
##
## Sample avg. posterior predictive distribution of y:
##          Median MAD_SD
## mean_PPD 0.1    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Not necessary at the moment

*Label variables so tables look nice label var SCDBradVidManipAll_mean “SCD (Mean) During Video” label var SCDBradSelfReport1_mean “SCD (Mean) While Answering Questions” label var emo “Self-Reported Immigration Beliefs” label var CellID “Brader Condition (6 cells)” label var anxcond “Anxiety Manipulation Dummy” label var storycond “Story Condition” label var interaction “Story X Anxiety” label variable age “Age” label variable race “Race” label variable income “Income” label variable education “Education” label variable ideology “Ideology” label variable anxietyvid “Anxiety Manipulation” label variable relaxvid “Relax Manipulation” label variable anxcond3 “Anxiety Condition” label variable immigration “Immigration DV”

Figure 2 Means of Skin Conductance Reactivity by Video Condition ciplot SCDBradVidManipAll_mean,by(anxcond3)

- Table 1 in paper
- (1) reg SCDBradSelfReport1_mean anxcond if anxcond3 ~=0
- (2) reg immigration storycond anxcond SCDBradSelfReport1_mean if anxcond3 ~=0

Figure 2

Want to obtain a confidence interval. Can do so by running a t.test(), but want to look into more effective ways.

Figure 3