Answer Key for Intro To Sparse Modelling

Elastic Net

```
### categorical architecture
elastic.cv.cat <- cv.glmnet(sim.x.training, y_cat.training, type.measure = "deviance",
    nfolds = 5, alpha = 0.5)
### remember that there's an intercept, so it's not exactly the same number
cor.test(coef(elastic.cv.cat)[2:10001, ], c(rep(0.7, 4), rep(0, 9996)))
predicted_y_elastic_cat <- predict(elastic.cv.cat, newx = sim.x.test)
cor.test(y_cat.test, predicted_y_elastic_cat)

### quantitative architecture
elastic.cv.quant <- cv.glmnet(sim.x.training, y_quant.training, type.measure = "deviance",
    nfolds = 5, alpha = 0.5)
### remember that there's an intercept, so it's not exactly the same number
cor.test(coef(elastic.cv.quant)[2:10001, ], c(rep(0.35, 10), rep(0, 9990)))
predicted_y_elastic_quant <- predict(elastic.cv.quant, newx = sim.x.test)
cor.test(y_quant.test, predicted_y_elastic_quant)

### polygenic architecture
elastic.cv.polygenic <- cv.glmnet(sim.x.training, y_polygenic.training, type.measure = "deviance",
    nfolds = 5, alpha = 0.5)
### remember that there's an intercept, so it's not exactly the same number
cor.test(coef(elastic.cv.polygenic)[2:10001, ], c(rep(0.1, 100), rep(0, 9900)))
predicted_y_elastic_polygenic <- predict(elastic.cv.polygenic, newx = sim.x.test)
cor.test(y_polygenic.test, predicted_y_elastic_polygenic)
```

Ridge

```
### categorical architecture
ridge.cv.cat <- cv.glmnet(sim.x.training, y_cat.training, type.measure = "deviance",
    nfolds = 5, alpha = 0)
### remember that there's an intercept, so it's not exactly the same number
cor.test(coef(ridge.cv.cat)[2:10001, ], c(rep(0.7, 4), rep(0, 9996)))
predicted_y_ridge_cat <- predict(ridge.cv.cat, newx = sim.x.test)
cor.test(y_cat.test, predicted_y_ridge_cat)

### quantitative architecture
ridge.cv.quant <- cv.glmnet(sim.x.training, y_quant.training, type.measure = "deviance",
    nfolds = 5, alpha = 0)
### remember that there's an intercept, so it's not exactly the same number
cor.test(coef(ridge.cv.quant)[2:10001, ], c(rep(0.35, 10), rep(0, 9990)))
predicted_y_ridge_quant <- predict(ridge.cv.quant, newx = sim.x.test)
cor.test(y_quant.test, predicted_y_ridge_quant)  ###no variation in the predicted phenotype

### polygenic architecture
ridge.cv.polygenic <- cv.glmnet(sim.x.training, y_polygenic.training, type.measure = "deviance",
    nfolds = 5, alpha = 0)
### remember that there's an intercept, so it's not exactly the same number
cor.test(coef(ridge.cv.polygenic)[2:10001, ], c(rep(0.1, 100), rep(0, 9900)))
```

```
predicted_y_ridge_polygenic <- predict(ridge.cv.polygenic, newx = sim.x.test)
cor.test(y_polygenic.test, predicted_y_ridge_polygenic)
```

Lasso

```
### categorical architecture
lasso.cv.cat <- cv.glmnet(sim.x.training, y_cat.training, type.measure = "deviance",
    nfolds = 5, alpha = 1)
### remember that there's an intercept, so it's not exactly the same number
cor.test(coef(lasso.cv.cat)[2:10001, ], c(rep(0.7, 4), rep(0, 9996)))
predicted_y_lasso_cat <- predict(lasso.cv.cat, newx = sim.x.test)
cor.test(y_cat.test, predicted_y_lasso_cat)


### quantitative architecture
lasso.cv.quant <- cv.glmnet(sim.x.training, y_quant.training, type.measure = "deviance",
    nfolds = 5, alpha = 1)
### remember that there's an intercept, so it's not exactly the same number
cor.test(coef(lasso.cv.quant)[2:10001, ], c(rep(0.35, 10), rep(0, 9990)))
predicted_y_lasso_quant <- predict(lasso.cv.quant, newx = sim.x.test)
cor.test(y_quant.test, predicted_y_lasso_quant)

### polygenic architecture
lasso.cv.polygenic <- cv.glmnet(sim.x.training, y_polygenic.training, type.measure = "deviance",
    nfolds = 5, alpha = 1)
### remember that there's an intercept, so it's not exactly the same number
cor.test(coef(lasso.cv.polygenic)[2:10001, ], c(rep(0.1, 100), rep(0, 9900)))
predicted_y_lasso_polygenic <- predict(lasso.cv.polygenic, newx = sim.x.test)
cor.test(y_polygenic.test, predicted_y_lasso_polygenic)
```