Scalable Data Engineering, Exercise 7

Henry Haustein

I don't know what happened to exercise 6, there aren't any tasks.

Task 1

(a) False. Most obviously SQL is not Turing complete.

(b) False. Functions do calculations and have a return value, procedures don't have a return value and can do data manipulations.

(c) False. An aggregate function calls the step function for every processed tuple and a final function on the end.

(d) False. Regression can do this but it's not limited to that.

(e) False. Overfitting is a real issue.

(f) False. We don't necessarily want good performance on the training set, we want it on the test set. Overfitting does not give that.

(g) False. In general this is not a good idea.

Task 2

With R this is pretty easy

```r
x = c(1, 2.5, 3, 4.25, 5)
y = c(4.5, 4, 2.5, 2, 1)
summary(lm(y ~ x))
```

Gives

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	5.5928	0.5247	10.66	0.0018
x	-0.8866	0.1523	-5.82	0.0101

Task 3

In SQL:

```
CREATE TABLE reg (size numeric, price numeric);

INSERT INTO reg (size, price) VALUES (1000, 275000), (2500, 370000), (800, 175000), (1900, 225000), (3000, 500000);

CREATE EXTENSION plpython3u;

CREATE OR REPLACE FUNCTION ols() RETURNS numeric[] AS
$$
  from sklearn.linear_model import LinearRegression

  X = []
  y = []

  rv = plpy.execute("SELECT * FROM reg")
  for row in rv:
    X.append([row["size"]])
    y.append(row["price"])

  model = LinearRegression().fit(X, y)
  return [*model.coef_, model.intercept]
$$
LANGUAGE plpython3u;

SELECT ols();
```