Project 1

This dataset is adapted from the World Health Organization on Strokes (it's based on real data but is NOT REAL). Use this dataset to answer the following questions and perform the following tasks. Feel free to add extra cells as needed, but follow the structure listed here and clearly identify where each question is answered. Please remove any superflous code.

Data Information

- `reg_to_vote`: 0 if no, 1 if yes.
- `age`: age of the patient in years.
- `hypertension`: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension.
- `heart_disease`: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease.
- `ever_married`: 0 if no, 1 if yes.
- `Residence_type`: 0 for Rural, 1 for Urban.
- `avg_glucose_level`: average glucose level in blood.
- `bmi`: body mass index.
- `smoking_status_smokes`, `smoking_status_formerly`: Whether or not the person smokes, or formerly smoked. If a person has 0's for both these columns, they never smoked.
- `stroke`: 1 if the patient had a stroke or 0 if not.
- `dog_owner`: 0 if no, 1 if yes.
- `er_visits`: number of recorded Emergency Room visits in lifetime.
- `racoons_to_fight`: number of racoons the patient belives they could fight off at once.
- `fast_food_budget_month`: amount (in US dollars) spent on fast food per month.

Part I: Logistic Regression

Build a logistic regression model to predict whether or not someone had a stroke based on **all** the other variables in the dataset.

1. Count the missing data per column, and remove rows with missing data (if any).
2. Use 10 fold cross validation for your model validation. Store both the train and test accuracies to check for overfitting. **Is the model overfit? How can you tell?** Z-score your continuous variables only.
3. After completing steps 1-3, fit another logistic regression model on ALL of the data (no model validation) using the same predictors as before, and put the coefficients into a dataframe called `coef`.
4. print out a confusion matrix for the model you made in part 3. **What does this confusion matrix tell you about your model? How can you tell?**

Part II: Data Exploration

The WHO has asked the following five questions, create **at least 1 ggplot graph** (using the above data + model when needed) to help answer each question, and **explicitly answer the question in a Markdown cell** below your graph. You may use other calculations to help support your answer but MUST pair it with a graph. Write your answer as if you were explaining it to a non-data scientist. You will be graded on the effectiveness and clarity of your graph, as well as the completeness, clarity, and correctness of your responses and justifications.

1. Do dog-ownders over 50 have a higher probability of stoke than non-dog owners who currently smoke? How can you tell?
2. What is the relationship between average blood glucose and BMI? Is the relationship between those two variables different for people who are and are not registered to vote? How can you tell?
3. Is your logistic regression model most accurate for people who make less than 30k, between 30-90k, or over 100k? Discuss the potential accuracy *and* ethical implications if your model *were* more accurate for different groups (you can use the full model from part I-3 to check accuracy).
4. Which of the following variables is the strongest predictor of having a stroke (owning a dog, residence type, marriage, being registered to vote)? How were you able to tell?

5. **Create a variable `er_visits_per_year` that calculates the # of visits to the ER that a person has had per year of life. Store this variable in your data frame (no need to include this variable in the previous logistic regression model). Is the # of ER visits per year different for stroke and non-stroke patients? How can you tell?**

PART I

In [73]:

```
import warnings
warnings.filterwarnings('ignore')


import pandas as pd
import numpy as np
from plotnine import *

from sklearn.linear_model import LogisticRegression # Logistic Regression Model
from sklearn.preprocessing import StandardScaler #Z-score variables
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import plot_confusion_matrix

from sklearn.model_selection import train_test_split # simple TT split cv
from sklearn.model_selection import KFold # k-fold cv
from sklearn.model_selection import LeaveOneOut #LOO cv
from sklearn.model_selection import cross_val_score # cross validation metrics
from sklearn.model_selection import cross_val_predict # cross validation metrics
```

In [114]:

```
#PART I
heart = pd.read_csv("https://raw.githubusercontent.com/cmparlettpelleriti/CPSC392ParlettPelleriti/master/Data/Proj1.csv")
heart.isnull().sum(axis=0)
stroke = heart.dropna()
stroke.head(50)
```

Out[114]:

	age	hypertension	heart_disease	ever_married	Residence_type	avg_glucose_level	bmi	stroke	smoking_status_smokes
0	60.0	1.0	0.0	0.0	1.0	73.00	25.2	0	1
1	4.0	0.0	0.0	0.0	0.0	110.15	17.1	0	0
2	77.0	0.0	0.0	1.0	1.0	68.38	27.8	0	0
3	37.0	0.0	0.0	1.0	1.0	95.08	30.1	0	0
4	44.0	0.0	0.0	0.0	0.0	103.78	40.9	0	1
5	77.0	0.0	0.0	1.0	1.0	99.78	29.5	0	0
6	56.0	0.0	0.0	1.0	0.0	156.18	25.8	0	0
7	53.0	0.0	0.0	1.0	1.0	126.35	24.9	0	0
8	37.0	0.0	0.0	1.0	0.0	74.29	24.6	0	1
9	39.0	0.0	0.0	1.0	0.0	73.07	33.4	0	0
10	33.0	0.0	0.0	1.0	0.0	73.20	28.9	0	0
11	71.0	0.0	1.0	1.0	1.0	215.72	32.4	0	1
12	26.0	0.0	0.0	1.0	1.0	116.38	22.3	0	0
13	42.0	0.0	0.0	0.0	1.0	84.43	30.5	0	0
14	9.0	0.0	0.0	0.0	0.0	77.67	19.4	0	0
15	25.0	0.0	0.0	0.0	0.0	65.77	24.7	0	0
16	68.0	0.0	1.0	1.0	0.0	193.45	31.0	0	1
17	52.0	0.0	0.0	1.0	0.0	191.66	25.0	0	1

	age	hypertension	heart_disease	ever_married	Residence_type	avg_glucose_level	bmi	stroke	smoking_status_smokes
17	52.0	0.0	0.0	1.0	0.0	191.66	25.0	0	1
18	30.0	0.0	0.0	1.0	1.0	95.94	29.8	0	0
19	74.0	0.0	0.0	1.0	0.0	94.67	27.2	0	0
20	15.0	0.0	0.0	0.0	1.0	95.86	18.1	0	0
21	34.0	0.0	0.0	0.0	1.0	60.36	26.5	0	0
22	4.0	0.0	0.0	0.0	1.0	81.87	17.2	0	0
23	74.0	1.0	1.0	1.0	0.0	93.62	19.7	0	0
24	56.0	1.0	0.0	1.0	0.0	97.37	32.3	1	0
25	35.0	0.0	0.0	1.0	1.0	205.97	29.5	0	0
26	60.0	0.0	0.0	1.0	0.0	103.17	26.4	0	1
27	42.0	0.0	0.0	1.0	1.0	74.80	50.6	0	0
29	43.0	0.0	0.0	1.0	0.0	81.77	23.8	0	1
31	76.0	0.0	0.0	1.0	0.0	221.80	34.5	1	0
32	50.0	0.0	0.0	1.0	0.0	119.77	31.6	0	0
33	53.0	0.0	0.0	1.0	1.0	74.66	30.0	0	1
34	78.0	0.0	0.0	1.0	1.0	58.88	24.3	0	0
35	53.0	1.0	0.0	1.0	1.0	202.66	34.1	0	0
36	71.0	0.0	0.0	1.0	1.0	134.65	35.9	0	1
37	8.0	0.0	0.0	0.0	1.0	88.02	16.0	0	0
38	15.0	0.0	0.0	0.0	0.0	114.53	30.5	0	0
39	36.0	0.0	0.0	0.0	1.0	77.12	27.6	0	0
40	50.0	1.0	0.0	1.0	0.0	220.36	29.5	1	1
41	34.0	0.0	0.0	0.0	1.0	67.66	33.0	0	0
42	78.0	0.0	0.0	1.0	0.0	56.34	29.6	0	1
43	1.0	0.0	0.0	0.0	0.0	88.11	20.3	0	0
44	70.0	1.0	0.0	1.0	1.0	65.98	29.1	0	0
45	59.0	0.0	0.0	1.0	0.0	79.18	30.3	0	0
46	69.0	1.0	0.0	0.0	1.0	96.52	26.1	1	0
47	24.0	0.0	0.0	0.0	0.0	72.06	20.5	0	0
48	24.0	0.0	0.0	0.0	0.0	123.89	24.1	0	0
49	7.0	0.0	0.0	0.0	1.0	56.32	16.8	0	0
50	38.0	0.0	0.0	1.0	1.0	162.30	23.6	0	1
51	22.0	0.0	0.0	0.0	0.0	74.99	33.9	0	0

In [75]:

```
predictors = ["reg_to_vote", "age", "hypertension", "heart_disease", "ever_married", "bmi
", "Residence_type","avg_glucose_level", "smoking_status_smokes","smoking_status_formerly
", "dog_owner", "er_visits", "raccoons_to_fight", "fast_food_budget_month","income_in_k"]
X = stroke[predictors]
y = stroke["stroke"]

# create k-fold object
kf = KFold(n_splits = 10)
kf.split(X)
lr = LogisticRegression() #create model

acc_train = []
acc_test = []
```

```
for train_indices, test_indices in kf.split(X):
    # Get your train/test for this fold
    X_train = X.iloc[train_indices]
    X_test  = X.iloc[test_indices]
    y_train = y.iloc[train_indices]
    y_test  = y.iloc[test_indices]
    #zscore
    zScore = StandardScaler()
    zScore.fit(X_train[["age", "avg_glucose_level", "bmi", "er_visits", "raccoons_to_fig
ht", "fast_food_budget_month","income_in_k"]])
    Z = zScore.transform(X_train[["age", "avg_glucose_level", "bmi", "er_visits", "racco
ons_to_fight", "fast_food_budget_month","income_in_k"]])
    X_train[["age", "avg_glucose_level", "bmi", "er_visits", "raccoons_to_fight", "fast_
food_budget_month","income_in_k"]] = Z
    zScore.fit(X_test[["age", "avg_glucose_level", "bmi", "er_visits", "raccoons_to_figh
t", "fast_food_budget_month","income_in_k"]])
    Z = zScore.transform(X_test[["age", "avg_glucose_level", "bmi", "er_visits", "raccoo
ns_to_fight", "fast_food_budget_month","income_in_k"]])
    X_test[["age", "avg_glucose_level", "bmi", "er_visits", "raccoons_to_fight", "fast_f
ood_budget_month","income_in_k"]] = Z
    #create model
    model = lr.fit(X_train, y_train)
    acc_test.append(accuracy_score(y_test, model.predict(X_test)))
    acc_train.append(accuracy_score(y_train, model.predict(X_train)))

#print overall acc
print("training acc:", acc_train, "testing acc:", acc_test)
```

```
training acc: [0.9603093991718104, 0.9613251035237128, 0.96109375, 0.9596875, 0.9603125,
0.959921875, 0.961171875, 0.961015625, 0.95984375, 0.959765625] testing acc: [0.962052002
8109627, 0.9501054111033029, 0.9556962025316456, 0.9683544303797469, 0.959915611814346, 0
.9634317862165963, 0.9556962025316456, 0.9528832630098453, 0.9648382559774965, 0.96694796
06188467]
```

In [76]:

```
acc_train2 = np.mean(acc_train)
acc_test2 = np.mean(acc_test)
print("training acc:", acc_train2, "testing acc:", acc_test2)
```

```
training acc: 0.9604447002695522 testing acc: 0.9599921126994435
```

1. **ANSWER HERE**
2. **ANSWER HERE**

Is the model overfit?

No the model is not overfit because the training and testing accuracy scores are very close to each other. A model is over fit when the training accrucary score is higher than the testing accuracy score.In general models should have a high accuracy score,because the accuracy score shows how how well the models predict correctly. The higher accuracy score means the model performs better.

In [77]:

```
#fit another logistic regression model on ALL of the data (no model validation) using the
same predictors as before, and put the coefficients into a dataframe called coef.

predictors = ["reg_to_vote", "age", "hypertension", "heart_disease", "ever_married", "bmi
", "Residence_type","avg_glucose_level", "smoking_status_smokes","smoking_status_formerly
", "dog_owner", "er_visits", "raccoons_to_fight", "fast_food_budget_month","income_in_k"]
X = stroke[predictors]
y = stroke["stroke"]
#z-score
zScore = StandardScaler()
zScore.fit(stroke[["age", "avg_glucose_level", "bmi", "er_visits", "raccoons_to_fight",
"fast_food_budget_month","income_in_k"]])
Z = zScore.transform(stroke[["age", "avg_glucose_level", "bmi", "er_visits", "raccoons_t
o_fight", "fast_food_budget_month","income_in_k"]])
stroke[["age", "avg_glucose_level", "bmi", "er_visits", "raccoons_to_fight", "fast_food_
budget_month","income_in_k"]] = Z
#create model
model2 = LogisticRegression(penalty = "none" )
model2.fit(X,y)
predictedvalues= model2.predict(X)
coef = pd.DataFrame({"Coefs": model2.coef_[0], "Names": predictors})
coef = coef.append({"Coefs": model2.intercept_[0], "Names": "intercept"}, ignore_index =
True)
coef
```

Out[77]:

	Coefs	Names
0	-0.692332	reg_to_vote
1	0.045149	age
2	0.579185	hypertension
3	0.626132	heart_disease
4	-0.430726	ever_married
5	-0.082502	bmi
6	-0.503679	Residence_type
7	0.002394	avg_glucose_level
8	0.138746	smoking_status_smokes
9	0.117918	smoking_status_formerly
10	-0.425749	dog_owner
11	-0.024501	er_visits
12	-0.088676	raccoons_to_fight
13	0.001772	fast_food_budget_month
14	-0.007058	income_in_k
15	-1.256294	intercept

In [78]:

```
#print out a confusion matrix for the model you made in part 3.

confusion_matrix(y,predictedvalues)
```

Out[78]:

```
array([[13636,    10],
       [  572,     4]])
```

What does this confusion matrix tell you about your model? How can you tell?

The number 13636 means that 13636 people were predicted to have a stroke and they did have a stroke. The number 10 means that 10 people were predicted to have a stroke but they did not have a stroke. The number 572 means that 572 people were predicted to not have a stroke but they did have a stroke. The number 4 menas that 4 people were preditced to not have a stroke and they did not have a stroke

In [93]:

```
accuracy_score(y, predictedvalues)
```

Out[93]:

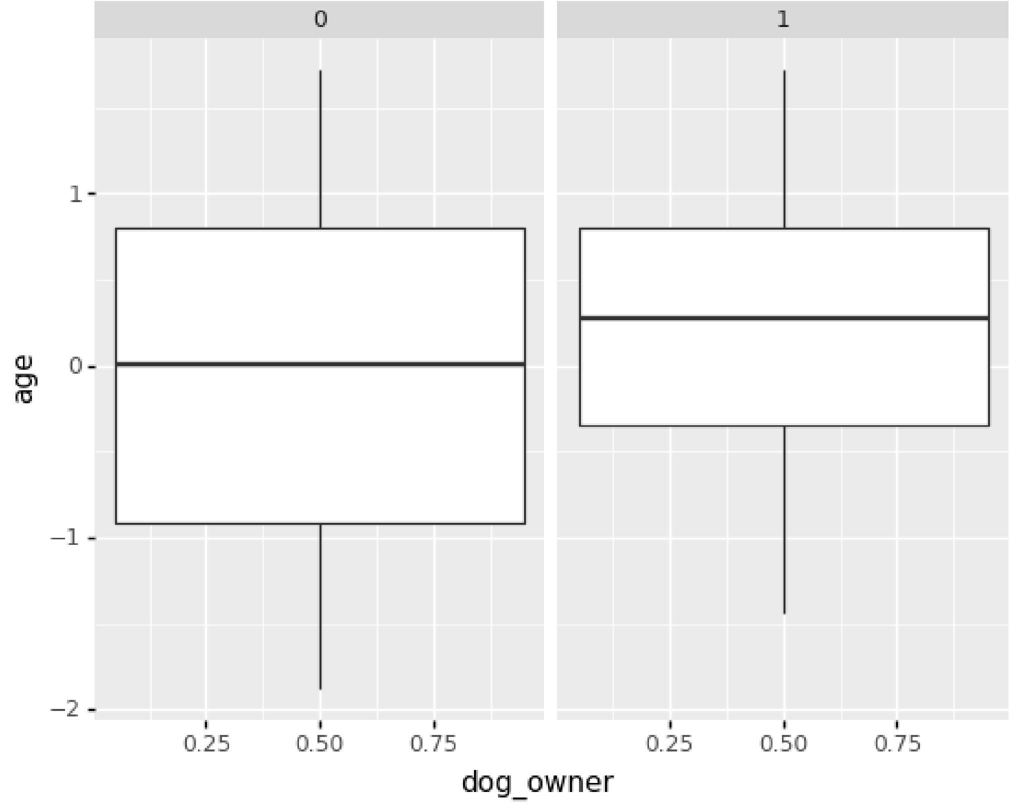
```
0.9590774855857123
```

PART II

In [60]:

```
# Do dog-ownders over 50 have a higher probability of stoke than non-dog owners who curre
ntly smoke? How can you tell?

(ggplot(stroke, aes(x = "dog_owner", y = "age")) + geom_boxplot() + facet_wrap("smoking_
status_smokes"))
```



Out[60]:

```
<ggplot: (8793865394083)>
```

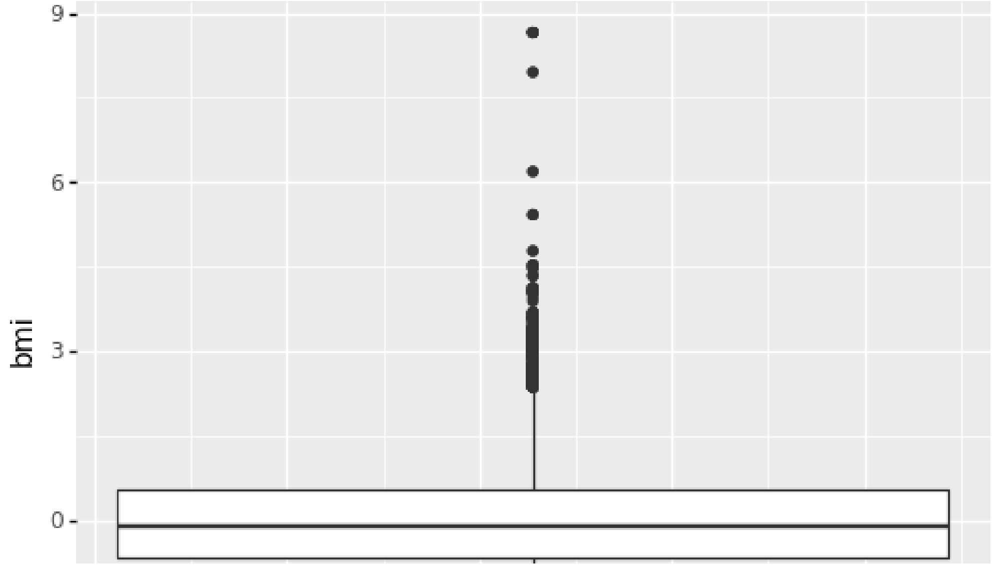
It seems like not matter if you are a dog owner or not, if you are over 50, you are more likely to have a stroke. Again it seems like not matter if you are a dog owner or not, if you smoke, you are more likely to have a stroke. So the probabilty of a dog owner over 50 having a stroke is the same as a non-dog owner who smokes having a stroke.

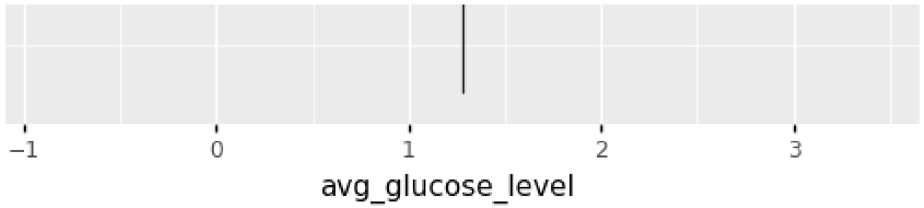
1. **DISCUSSION + ANSWERE HERE**

In [59]:

```
# What is the relationship between average blood glucose and BMI?

(ggplot(stroke, aes(x = "avg_glucose_level", y = "bmi")) + geom_boxplot())
```





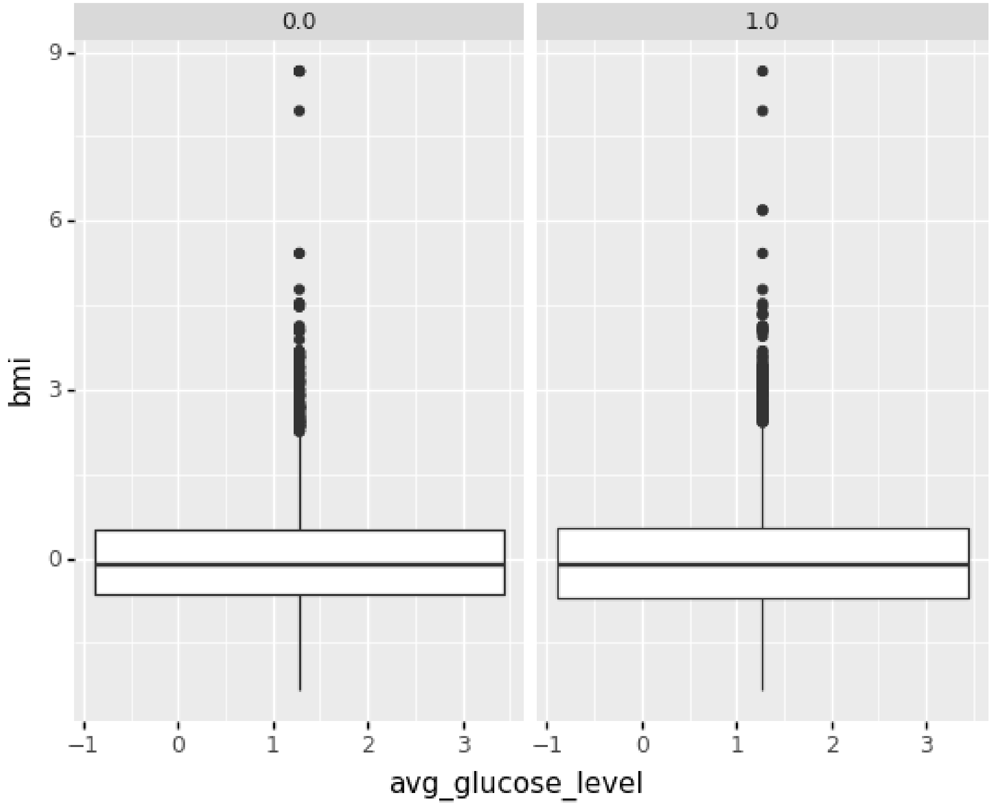
Out[59]:

```
<ggplot: (8793835262874)>
```

In [98]:

```
# Is the relationship between those two variables different for people who are and are no
t registered to vote? How can you tell?

(ggplot(stroke, aes(x = "avg_glucose_level", y = "bmi")) + geom_boxplot() + facet_wrap("
~reg_to_vote"))
```



Out[98]:

```
<ggplot: (8793835507167)>
```

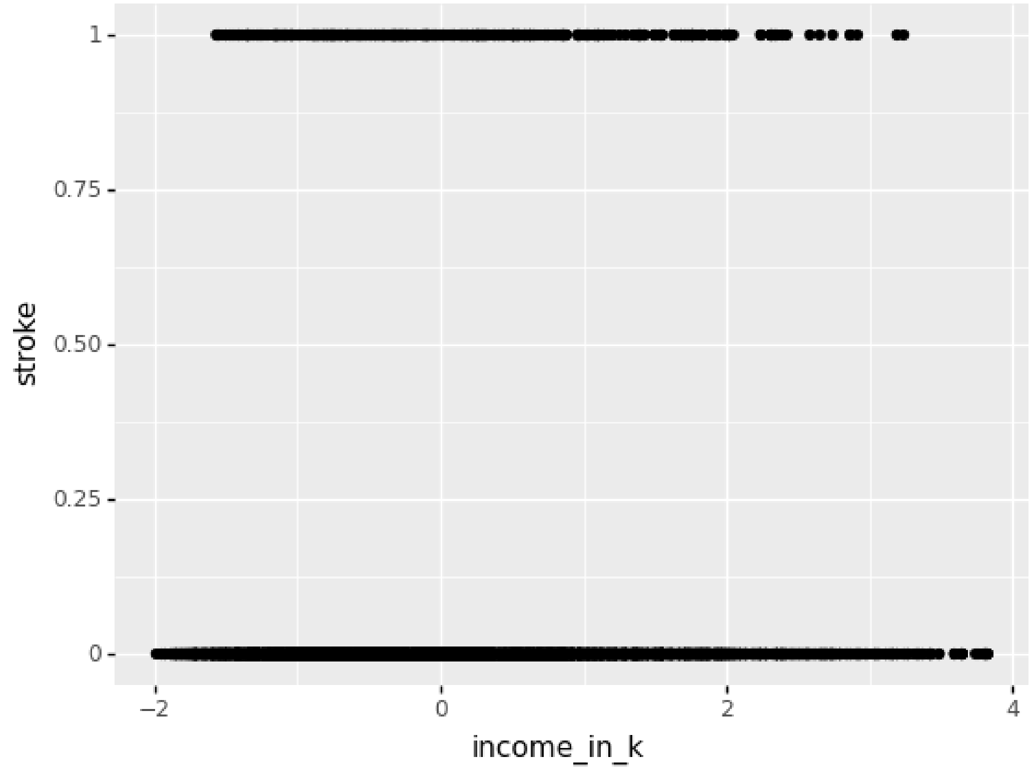
The first boxplot has a positive skew, meaning the data constitute higher frequency of high valued scores and the mean is greater than the median. This means if someone has a high glucose level, they are more likely to have a higher BMI. No, the relationship doesn't change if they are registered to vote or not. As seen by the two boxplots, their medians are about the same. The 25% and 75% quartile are the same length and their max/min are the same length. They also have very similar outlier points. Because of this, the relationship does not change.

1. **DISCUSSION + ANSWERE HERE**

In [92]:

```
# Is your logistic regression model most accurate for people who make less than 30k, betw
een 30-90k, or over 100k? Discuss the potential accuracy and ethical implications if your
model were more accurate for different groups (you can use the full model from part I-3 t
o check accuracy).

ggplot(stroke, aes(x="income_in_k", y="stroke")) + geom_point(stat = "identity")
```



Out[92]:

```
<ggplot: (8793835506014)>
```

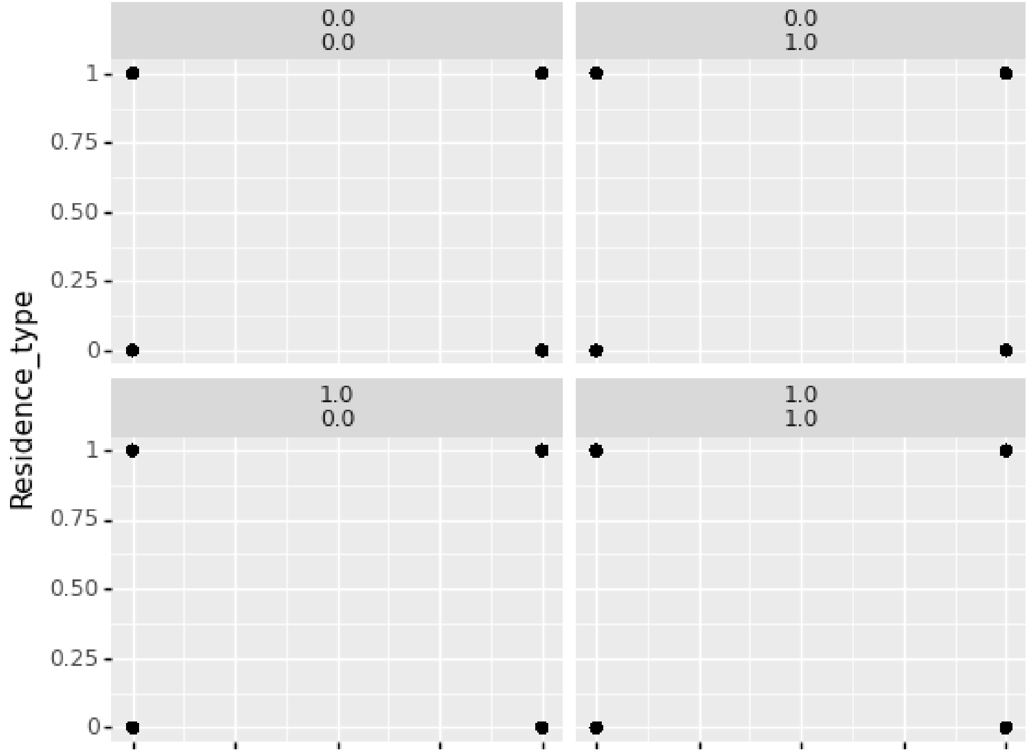
According to the coef chart, for every 1 increase in income_in_k, there is a decrease of -.007 in having a stroke. So the lower the income, the higher chnace of a stroke. The model is doing well in predicting since the accuracy score .959. We want a high accuracy score.The ethical implications are sad. If you have a low income, you are more liekly yo have a stroke. You are less likely to be able to pay for the treatments, and that is not fair.

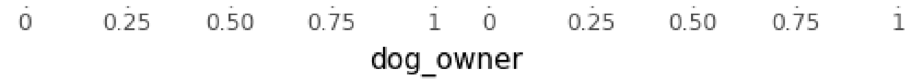
1. **DISCUSSION + ANSWERE HERE**

In [101]:

```
#Which of the following variables is the strongest predictor of having a stroke (owning a
dog, residence type, marriage, being registered to vote)? How were you able to tell?

(ggplot(stroke, aes(x = "dog_owner", y = "Residence_type")) + geom_point() + facet_wrap(
"~ ever_married + reg_to_vote"))
```





Out[101]:

```
<ggplot: (8793860826799)>
```

The variable which is the strongest predictor of having a stroke is owning a dog. This is because the owning a dog has the standardized coefficient with the smallest absolute value. The owning a dog coef is -0.425749 which is 0.425749 when absoluted. The being registered to vote coef is -.69233 which is .69233 when absoluted. The marraige coef is -0.430726 which is 0.430726 when absoluted. Finally residence type coef is -0.503679 which is 0.503679 when absoluted. This measure suggests that being registered to vote is the most important independent variable in the regression model, since it is the standardized coefficient with the largest absolute value. While this measure suggests that owning a dog is the strongest variable in the regression model, since it is the standardized coefficient with the smallest absolute value.

1. **DISCUSSION + ANSWERE HERE**

In [121]:

```
# Create a variable er_visits_per_year that calculates the # of visits to the ER that a p
erson has had per year of life. Store this variable in your data frame (no need to includ
e this variable in the previous logistic regression model).

stroke["er_visit_per_year"] = stroke["er_visits"]/(stroke["age"])
stroke.head(25)
```

Out[121]:

	age	hypertension	heart_disease	ever_married	Residence_type	avg_glucose_level	bmi	stroke	smoking_status_smokes
0	60.0	1.0	0.0	0.0	1.0	73.00	25.2	0	1
1	4.0	0.0	0.0	0.0	0.0	110.15	17.1	0	0
2	77.0	0.0	0.0	1.0	1.0	68.38	27.8	0	0
3	37.0	0.0	0.0	1.0	1.0	95.08	30.1	0	0
4	44.0	0.0	0.0	0.0	0.0	103.78	40.9	0	1
5	77.0	0.0	0.0	1.0	1.0	99.78	29.5	0	0
6	56.0	0.0	0.0	1.0	0.0	156.18	25.8	0	0
7	53.0	0.0	0.0	1.0	1.0	126.35	24.9	0	0
8	37.0	0.0	0.0	1.0	0.0	74.29	24.6	0	1
9	39.0	0.0	0.0	1.0	0.0	73.07	33.4	0	0
10	33.0	0.0	0.0	1.0	0.0	73.20	28.9	0	0
11	71.0	0.0	1.0	1.0	1.0	215.72	32.4	0	1
12	26.0	0.0	0.0	1.0	1.0	116.38	22.3	0	0
13	42.0	0.0	0.0	0.0	1.0	84.43	30.5	0	0
14	9.0	0.0	0.0	0.0	0.0	77.67	19.4	0	0
15	25.0	0.0	0.0	0.0	0.0	65.77	24.7	0	0
16	68.0	0.0	1.0	1.0	0.0	193.45	31.0	0	1
17	52.0	0.0	0.0	1.0	0.0	191.66	25.0	0	1
18	30.0	0.0	0.0	1.0	1.0	95.94	29.8	0	0
19	74.0	0.0	0.0	1.0	0.0	94.67	27.2	0	0
20	15.0	0.0	0.0	0.0	1.0	95.86	18.1	0	0
21	34.0	0.0	0.0	0.0	1.0	60.36	26.5	0	0
22	4.0	0.0	0.0	0.0	1.0	81.87	17.2	0	0
23	71.0	1.0	1.0	1.0	0.0	93.60	19.7	0	0

	age	hypertension	heart_disease	ever_married	Residence_type	avg_glucose_level	bmi	stroke	smoking_status_smokes
23	74.0	1.0	1.0	1.0	0.0	93.62	19.7	0	0
24	56.0	1.0	0.0	1.0	0.0	97.37	32.3	1	0

Is the of ER visits per year different for stroke and non-stroke patients? How can you tell?

The number of ER visit per year is different for stroke and non-stroke patients. I can tell because the first person (#0) on the chart did not have a stroke and he has .155 visits to the er per year. The 24 person had a stroke and has .161 visits to the er per year. While the second person did not have a stroke, but has 1.25 visits to the er per year. It depends on the person's age and general health conditions, not on whether they have had a stroke or not. You can see this because the 24th person had storke but only visits the er .161 times a year versus the 2nd person, who has not had a stroke, but visits the er 1.25 times a year.

1. **DISCUSSION + ANSWERE HERE**