Midterm Exam

Hao Shen

11/7/2020

Instruction

This is your midterm exam that you are expected to work on it alone. You may NOT discuss any of the content of your exam with anyone except your instructor. This includes text, chat, email and other online forums. We expect you to respect and follow the GRS Academic and Professional Conduct Code.

Although you may NOT ask anyone directly, you are allowed to use external resources such as R codes on the Internet. If you do use someone's code, please make sure you clearly cite the origin of the code.

When you finish, please compile and submit the PDF file and the link to the GitHub repository that contains the entire analysis.

Introduction

In this exam, you will act as both the client and the consultant for the data that you collected in the data collection exercise (20pts). Please note that you are not allowed to change the data. The goal of this exam is to demonstrate your ability to perform the statistical analysis that you learned in this class so far. It is important to note that significance of the analysis is not the main goal of this exam but the focus is on the appropriateness of your approaches.

Data Description (10pts)

Please explain what your data is about and what the comparison of interest is. In the process, please make sure to demonstrate that you can load your data properly into R.

Lamb skewers are one of my favorite midnight snack. Near my home, there are nearly ten shops selling lamb skewers, and I often visit them after classes in the evening. Each store's skewers has its own characteristics, tastes and prices. However, what I am curious about here, or **the comparison of interest**, is which store can provide more meat under the same price?

For this, I started an experiment. In order to avoid waste, I only go to one store every day, and for each store I only orders 5 signature lamb skewers. For each lamb skewer, I collect the following data:

```
ls_raw=read.csv('lamb_skewers.csv',T)
ls_raw[1:2,]%>%
  kable('simple',align='c')
```

restaurant	serial	weight_total	weight_pole	lengh	count_fat	count_lean	price_unit
1	1	11.0	1.5	116	2	2	4.09
1	2	9.5	1.0	94	2	2	4.09

As shown in the table, there are 7 variables for each lamb skewer:

- **restaurant**: codes of different restaurant.
- **serial**: code of the lamb skewers of each restaurant.

- **weight_total**: the total weight of a skewer of meat, measured with an electronic scale with an accuracy of 0.5g.
- **weight_pole**: The weight of the skewer left after eating, measured with an electronic scale with an accuracy of 0.5g.
- **length**: the length of the skewers, measured with a ruler with an accuracy of 1 mm.
- **count_fat**: The number of fat on a skewer of meat, which I don't like to eat.
- **count_lean**: The number of lean meat on a skewer of meat, which is my favorite.
- **price_unit**: The unit price of each skewer of meat, calculated by dividing the total price (measured in CNY) by the number of strings.

Note: As you seen in the table, the column name of **length** has been mistyped as 'lengh', so we need to fix it.

```
colnames(ls_raw)[5]='length'
```

Since the comparison of interest is 'at the same price, which store can provide more meat', we need to create a new value **cost_effective** and it can be calculated as follow:

$$cost\_effective = \frac{weight\_total - weight\_pole}{price\_unit}$$

Here, the numerator of $weight\_total - weight\_pole$ represents the net weight of meat on each skewer, which eliminate the influence of weight variation of skewers. Then we divide it by **price_unit** to eliminate their price difference, so that we obtain the **cost_effective** which represents the comparison of interest.

Besides, since the **serial** only represents the code of the lamb skewers of each restaurant and our comparison of interest is about the weights between restaurants, we choose to ignore it in following analysis.

```
ls=ls_raw%>%
  mutate(cost_effective=(weight_total-weight_pole)/price_unit)%>%
  dplyr::select(-(serial:weight_pole),-price_unit)
ls$restaurant=as.factor(ls$restaurant)
ls[1:2,]%>%
  kable('simple',align='c')
```

restaurant	length	count_fat	count_lean	cost_effective
1	116	2	2	2.322738
1	94	2	2	2.078240

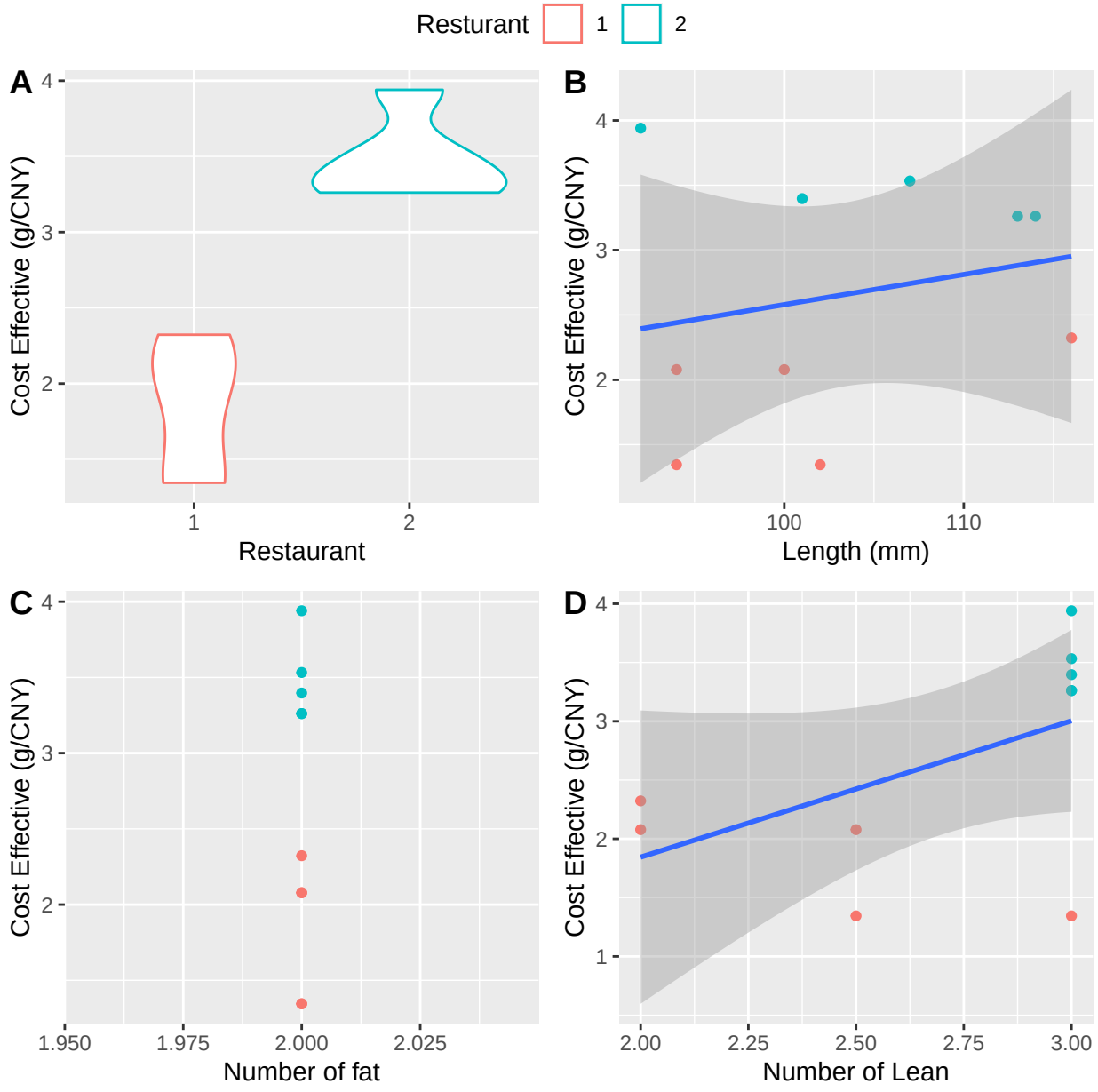
EDA (10pts)

Please create one (maybe two) figure(s) that highlights the contrast of interest. Make sure you think ahead and match your figure with the analysis. For example, if your model requires you to take a log, make sure you take log in the figure as well.

Firstly, let's take a look at the distribution of outcome and its relationship to variables.

```
ggarrange(ggplot(ls)+
            geom_violin(aes(restaurant,cost_effective,color=restaurant))+
            labs(x='Restaurant',y='Cost Effective (g/CNY)')+
            guides(color=guide_legend('Resturant')),
          ggplot(ls)+
            geom_point(aes(length,cost_effective,color=restaurant))+
            geom_smooth(aes(length,cost_effective),method='lm',formula=y~x)+
            labs(x='Length (mm)',y='Cost Effective (g/CNY)'),
          ggplot(ls)+
            geom_point(aes(count_fat,cost_effective,color=restaurant))+
            geom_smooth(aes(count_fat,cost_effective),method='lm',formula=y~x)+
```

```
  labs(x='Number of fat',y='Cost Effective (g/CNY)'),
ggplot(ls)+
  geom_point(aes(count_lean,cost_effective,color=restaurant))+
  geom_smooth(aes(count_lean,cost_effective),method='lm',formula=y~x)+
  labs(x='Number of Lean',y='Cost Effective (g/CNY)'),
ncol=2,nrow=2,common.legend=T,
labels=c('A','B','C','D')
)
```



From the plot A we can find that the cost effective of restaurant 2 is significantly larger than restaurant 1. Plot B and D shows that the length and the number of lean has a positive correlation with cost effective. As for Plot C, all observations have a value of 2 which means all skewers I received from 2 restaurants have identical 2 fats each. Since there is no variation against cost effective, we will choose to drop it in further regression step.

Then, let's explore the correlation between variables.

```
ls%>%
  dplyr::select(cost_effective,length,count_lean)%>%
  cor()%>%
  kable('simple',align='c')
```

	cost_effective	length	count_lean
cost_effective	1.0000000	0.2188383	0.5213965
length	0.2188383	1.0000000	0.0865244
count_lean	0.5213965	0.0865244	1.0000000

As the table shows, the correlation between remaining variables **length** and **count_lean** is low. And the correlation between outcome and variables are all positive.

Power Analysis (10pts)

Please perform power analysis on the project. Use 80% power, the sample size you used and infer the level of effect size you will be able to detect. Discuss whether your sample size was enough for the problem at hand. Please note that method of power analysis should match the analysis. Also, please clearly state why you should NOT use the effect size from the fitted model.

Since the comparison of interest **cost_effective** is to compares two groups of cases (Restaurant 1 and Restaurant 2) on one variable, we choose two independent samples t-test with two sided alternative hypothesis to calculate its Cohen's d Effect Size.

```
pwr.t.test(n=nrow(ls)/length(unique(ls$restaurant)),
           sig.level=0.05,
           power=0.8,
           type="two.sample",
           alternative='two.sided'
           )
```

```
##
##      Two-sample t test power calculation
##
##               n = 5
##               d = 2.024439
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

And from the result, the effect size is 2.024439, which is so large that could cause Type-M error.

In order to figure out the sample size we need for the lamb skewers problem, a medium Cohen's d Effect Size1 0.5, which is conceived as one large enough to be visible to the naked eye, is set for calculation.

```
pwr.t.test(d=0.5,
           sig.level=0.05,
           power=0.8,
           type="two.sample",
           alternative='two.sided'
           )
```

```
##
##      Two-sample t test power calculation
##
##               n = 63.76561
##               d = 0.5
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

The result shows at least 64 observations for each resturant is required, which is extremely larger than existing data set with 5 observations each. This is to say current modeling is not convincing.

Modeling (10pts)

Please pick a regression model that best fits your data and fit your model. Please make sure you describe why you decide to choose the model. Also, if you are using GLM, make sure you explain your choice of link function as well.

Since there is no large correlation between variables, we can first try to fit a simple linear regression.

```
(lm_ls=lm(cost_effective~restaurant+length+count_lean,ls))%>%
  summary()
```

```
##
## Call:
## lm(formula = cost_effective ~ restaurant + length + count_lean,
##     data = ls)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44273 -0.16378 -0.02539  0.17978  0.38002
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.619491   1.677153   2.754 0.033100 *
## restaurant2  2.211945   0.314283   7.038 0.000411 ***
## length      -0.006114   0.012451  -0.491 0.640825
## count_lean  -0.902908   0.381747  -2.365 0.055886 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3154 on 6 degrees of freedom
## Multiple R-squared:  0.9246, Adjusted R-squared:  0.8869
## F-statistic: 24.53 on 3 and 6 DF,  p-value: 0.0009102
```

As the model summary shows, the fit is pretty good since the adjusted R^2 is 0.8869 near to 1, but the p-value of **length** is too large. So, I believe we can improve the fit by dropping some variables.

```
(lm2_ls=step(lm_ls))%>%
  summary()
```

```
## Start:  AIC=-20.19
## cost_effective ~ restaurant + length + count_lean
##
##              Df Sum of Sq    RSS      AIC
```

```
## - length      1    0.0240 0.6207 -21.7951
## <none>                    0.5967 -20.1891
## - count_lean  1    0.5563 1.1531 -15.6017
## - restaurant  1    4.9263 5.5230   0.0633
##
## Step:  AIC=-21.8
## cost_effective ~ restaurant + count_lean
##
##              Df Sum of Sq    RSS      AIC
## <none>                     0.6207 -21.7951
## - count_lean  1    0.5337 1.1544 -17.5897
## - restaurant  1    5.1430 5.7637  -1.5101
```

```
##
## Call:
## lm(formula = cost_effective ~ restaurant + count_lean, data = ls)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4017 -0.1892 -0.0233  0.1184  0.4620
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.9294     0.8645   4.545 0.002651 **
## restaurant2   2.1684     0.2847   7.616 0.000125 ***
## count_lean   -0.8732     0.3559  -2.453 0.043884 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2978 on 7 degrees of freedom
## Multiple R-squared:  0.9216, Adjusted R-squared:  0.8992
## F-statistic: 41.13 on 2 and 7 DF,  p-value: 0.000135
```

The result shows after dropping **length**, the residual standard error decreases from 0.3154 to o.2978, and the adjusted R^2 also increases from 0.8992.

Then, since maybe the change from Restaurant 1 to Restaurant 2 do not have a fixed effect but actually a random effect to the **cost__effective**, we can also try two linear mixed-effects model, one with **length** and the other without to data.

```
(lmer_ls=lmer(cost_effective~(1|restaurant)+length+count_lean,ls))%>%
  summary()
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: cost_effective ~ (1 | restaurant) + length + count_lean
##    Data: ls
##
## REML criterion at convergence: 16.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4340 -0.4995 -0.1166  0.5766  1.2548
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  restaurant (Intercept) 2.39696  1.5482
```

```
##  Residual               0.09945  0.3154
## Number of obs: 10, groups:  restaurant, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  5.563363   2.078199   2.677
## length      -0.005616   0.012441  -0.451
## count_lean  -0.861948   0.379543  -2.271
##
## Correlation of Fixed Effects:
##            (Intr) length
## length     -0.695
## count_lean -0.589  0.155
```

```
(lmer2_ls=lmer(cost_effective~(1|restaurant)+count_lean,ls))%>%
  summary()
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: cost_effective ~ (1 | restaurant) + count_lean
##    Data: ls
##
## REML criterion at convergence: 10.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.3882 -0.6100 -0.1135  0.4191  1.5788
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  restaurant (Intercept) 2.31055  1.5200
##  Residual               0.08867  0.2978
## Number of obs: 10, groups:  restaurant, 2
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   4.9190     1.4417   3.412
## count_lean   -0.8382     0.3542  -2.366
##
## Correlation of Fixed Effects:
##            (Intr)
## count_lean -0.663
```

And using AIC to determine which model is better.

```
AIC(lm_ls,lm2_ls,lmer_ls,lmer2_ls)%>%
  kable('simple',align='c')
```

	df	AIC
lm_ls	5	10.189637
lm2_ls	4	8.583677
lmer_ls	5	26.930346
lmer2_ls	4	18.153398

As the the AIC shows, the second one, the improved simple linear model is the best one with the lowest AIC

of 8.583677.

Validation (10pts)

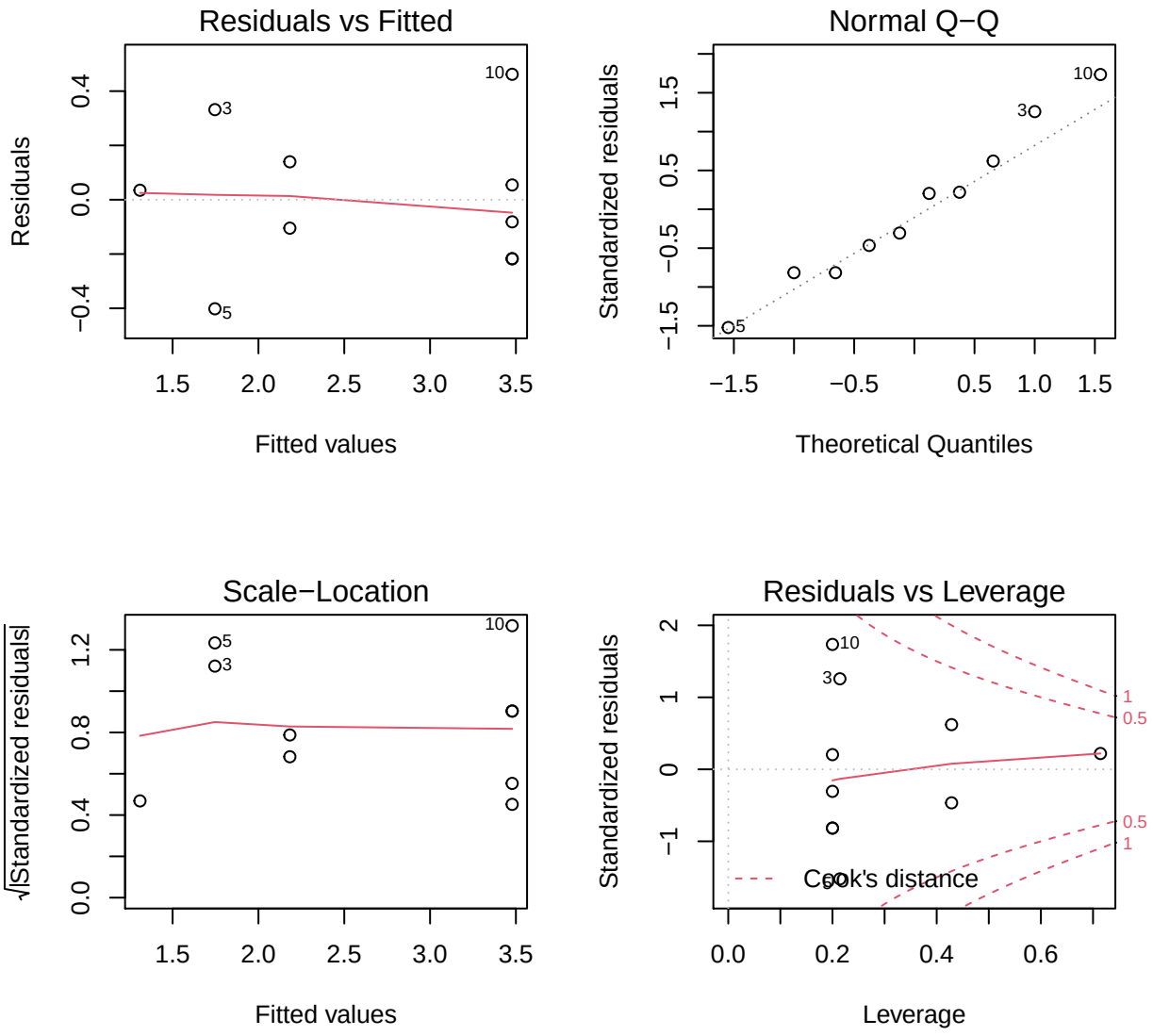
Please perform a necessary validation and argue why your choice of the model is appropriate.

Since we finally fitted a simple linear model for the data set, we need to check four things:

- The residuals are independent from variables.
- The residuals follow a normal distribution.
- The residuals have equal-variance.
- There is no outliers.

And we use 4 plots to check them.

```
par(mfrow=c(2,2))
plot(lm2_ls)
```



As the Residuals vs Fitted plot shows the red line is approximately parallel to X-axis, which means residuals are independent from variables. The Normal Q-Q plot and Scale-Location plot tell us that the residual is normal distributed with equal-variance respectively. And the Residuals vs Leverage plot indicate that there is no outliers in the regression.

However, it should be noticed that from power analysis, we found the sample size is not large enough to handle lamb skewers problem. The model we fitted here is not convincing.

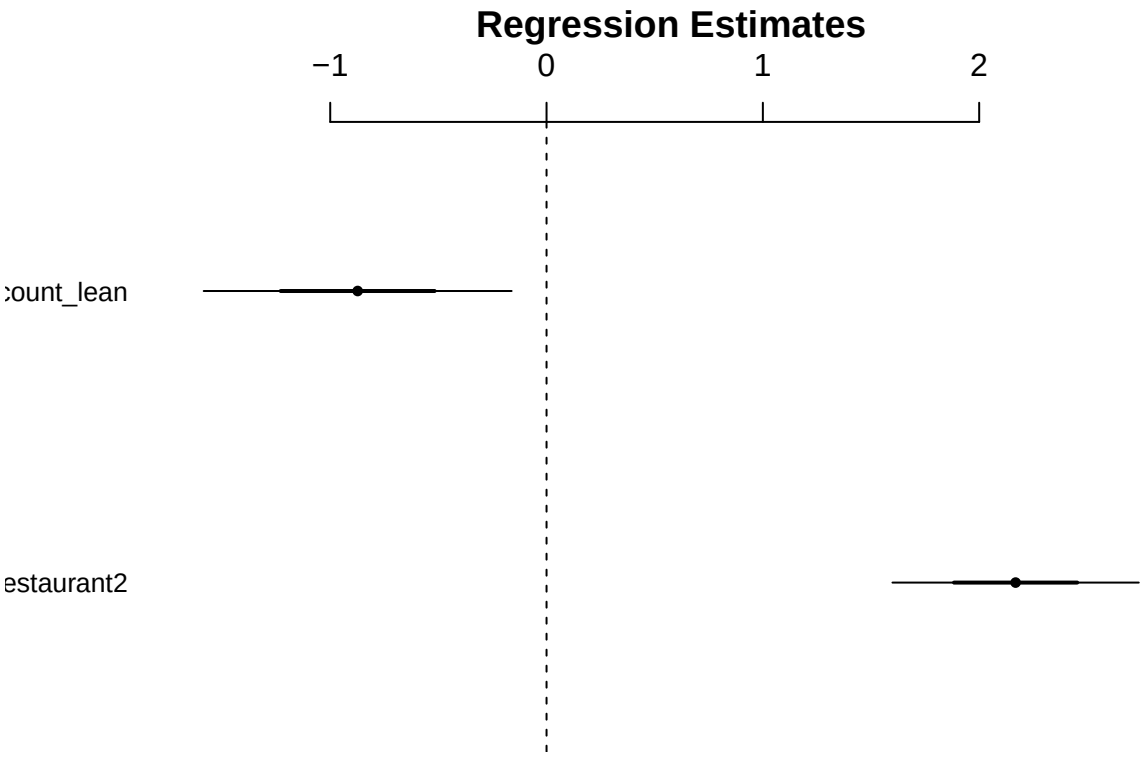
Inference (10pts)

Based on the result so far please perform statistical inference to compare the comparison of interest.

```
confint(lm2_ls)%>%
  kable('simple',align='c')
```

	2.5 %	97.5 %
(Intercept)	1.885226	5.9736631
restaurant2	1.495173	2.8417191
count_lean	-1.714801	-0.0316185

```
coefplot(lm2_ls)
```



Discussion (10pts)

Please clearly state your conclusion and the implication of the result.

In conclusion the simple linear model with variables of **restaurant** and **count_lean** is the best model for

the comparison of interest (under the same price, which restaurant can provide more meat), since it has the lowest AIC.

```
coef(lm2_ls)%>%
  kable('simple',align='c',col.names=NULL)
```

(Intercept)	3.9294446
restaurant2	2.1684460
count_lean	-0.8732099

And the final regression formula of the model is as follow:

$$cost\_effective = 3.93 - 0.87 * count\_lean + 2.17 * restaurant2$$

- The (Intercept) coefficient of 3.92 can be interpreted as the average **cost_effective** of restaurant 1 with zero lean on each skewer is 3.92.
- The count_lean coefficient of -0.87 can be interpreted as the average **cost_effective** of the same restaurant will decrease by 0.87 if they put 1 more lean on each skewer, which makes sense in real life as the unit price of lean is far more higher than that of fat.
- The restaurant2 coefficient of 2.17 can be interpreted as the average **cost_effective** of restaurant 2 is 2.17 higher than restaurant 1, which means, in general, the restaurant 2 can provide me with more meat than restaurant 2 under the same price. It indicates that I should go to restaurant 2 in the future.

However, above regression models are all not convincing because in the power analysis, it shows we need at least 64 skewers for each restaurant to fitted a model with medium effect size, but actually in this data set, there is only 5 skewers for each restaurant.

Besides, the linear mixed-effects models do not show better results than simple linear model, since there are only two group for random effects which is too small.

Limitations and future opportunity. (10pts)

Please list concerns about your analysis. Also, please state how you might go about fixing the problem in your future study.

- The sample size is too small to give a convincing conclusion, which should be increased in the future.
- The **count_fat** variable has been dropped at beginning since all of them are 2. It should be included to indicate more things in the future with more observations.
- The group amount of restaurants is small, and it should include more to compare all of restaurants near my home so that I can find a better one.
- The outcome of **cost_effective** is measured in g/CNY, which as displayed in regression is almost unrelated to the length of skewers. It should include some variables about weights, like the weight of lean and fat in the future.

Comments or questions

If you have any comments or questions, please write them here.

Actually, I have a trouble when using kable function to output matrix with a column name of '(Intercept)'. The R Markdown console shows it cannot be compiled with LaTex for the existence of '(' and ')'. I had to transform it to data.frame to fix it. In addition, I believe if there is a chance to update my data set the performance will be better.

Reference

[1] Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Hillsdale,NJ: Lawrence Erlbaum.