РЕГРЕССИЯ и КЛАССИФИКАЦИЯ

Регрессия

## Общие понятия и методы оценивания

<http://datascientist.one/machine-learning-algorithms-r-python/>

<http://www.science.smith.edu/~jcrouser/SDS293/labs/2016/>

набор питонов

https://machinelearningmastery.com/category/deep-learning-time-series/

# Однофакторный линейный регрессионный анализ в R

library(ISLR)

library(ggvis)

library(dplyr)

**Подготовка данных**

Работаем с массивом Auto из пакета ISLR.

# Загружаем массив Auto в пространство пользователя

data(Auto)

# Структура массива

str(Auto)

# Конвертируем миль (1.609 км) на галлон (3.79 л) в км на литр

Auto$kpl <- Auto$mpg \* 1.609 / 3.79

# Построение модели

Построим однофакторную регрессионную модель, чтобы выявлить, как мощность двигателя влияет на пробег на километр топлива, и ответим на следующие вопросы:

1. Существует ли взаимосвязь между мощностью и пробегом.
2. Насколько сильна взаимосвязь между мощностью и пробегом.
3. Взаимосвязь между мощностью и пробегом прямая или обратная.
4. Каково прогнозное значение пробега на километр при мощности 98 лошадиных сил. Каковы 95% доверительный интервал и интервал предсказания.

# Задаем модель

autolm <- lm(kpl ~ horsepower, data = Auto)

# Результаты анализа

summary(autolm)

Коэффициент horsepower составляет -0.067, т.е. при увеличении мощности автомобиля на одну лошадиную силу пробег снижается на 0.07 км на литр топлива. Наличие этой взаимосвязи подтверждается p-значением t-статистики: t равен -24.5, а вероятность получить такое значение t при независимых переменных (т.е. p-значение) стремится к нулю (<2e-16).

# Прогнозирование

# Прогнозируем пробег при мощности 98 л.с.

# Доверительный интервал

confint <- predict(autolm,

newdata = data.frame(horsepower = 98),

interval = 'confidence')

confint

## fit lwr upr

## 1 10.39 10.18 10.6

Если мы возьмём несколько разных автомобилей с мощностью двигателя 98 л.с., то средний пробег этих машин на литр топлива с вероятностью 95% попадёт в интервал от 10.2 до 10.6 км.

# Интервал предсказания

predint <- predict(autolm,

newdata = data.frame(horsepower = 98),

interval = 'prediction')

predint

## fit lwr upr

## 1 10.39 6.287 14.49

Если мы возьмём какой-то конкретный автомобиль с мощностью двигателя 98 л.с., то его пробег на литр топлива с вероятностью 95% попадёт в интервал от 6.3 до 14.5 км.

Auto %>%

ggvis(~horsepower, ~kpl) %>%

layer\_points() %>%

layer\_model\_predictions(model = 'lm', se = T)

# Диагностика

После построения регрессионной модели имеет смысл проверить наличие следующих проблем:

1. Нелинейная взаимосвязь между независимыми и зависимой переменными.
2. Корреляция остатков между собой.
3. Непостоянная дисперсия остатков (гетероскедастичность).
4. Наличие выбросов.
5. Избыточное влияние отдельных наблюдений.
6. Коллинеарность предикатов.

par(mfrow = c(2, 3))

plot(autolm, which = 1:6, caption =

list("Прогноз и остатки", "Квантили", "Разброс-Положение",

"Расстояние Кука", "Влияние и остатки",

expression("Влияние и Кук " \* h[ii] / (1 - h[ii]))))



# Нелинейность взаимосвязи

data.frame(predict = predict(autolm),

resids = residuals(autolm)) %>%

ggvis(~predict, ~resids) %>%

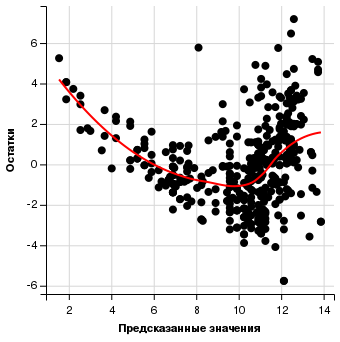
layer\_points() %>%

add\_axis("x", title = "Предсказанные значения") %>%

add\_axis("y", title = "Остатки") %>%

layer\_smooths(stroke := 'red')

<https://screenshots.firefox.com/AF3y8dXP9xuQmqxt/null>



Наблюдается некоторая нелинейная зависимость между остатками и предсказанными значениями. Имеет смысл сделать нелинейное преобразование элементов модели (логарифм, квадратный корень, квадрат).

# Корреляция остатков

data.frame(id = seq\_len(dim(Auto)[1]),

resids = residuals(autolm)) %>%

ggvis(~id, ~resids) %>%

layer\_paths() %>%

add\_axis("x", title = "Порядковый номер наблюдения") %>%

add\_axis("y", title = "Остатки")

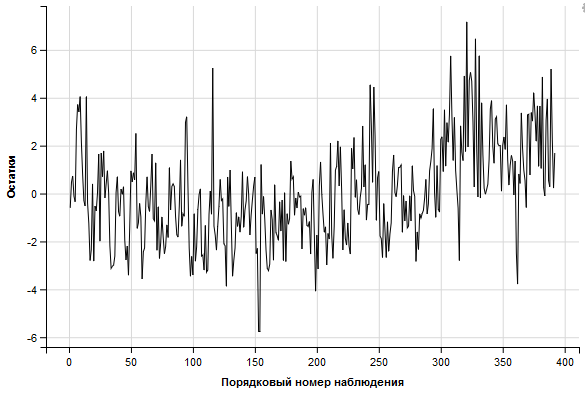


график не выглядит таким, что по нему на основе предыдущего наблюдения можно предсказать следующее. Делаем вывод, что корреляции остатков нет.

# Непостоянная дисперсия остатков

data.frame(predict = predict(autolm),

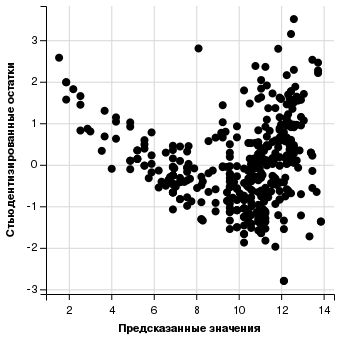
rstuds = rstudent(autolm)) %>%

ggvis(~predict, ~rstuds) %>%

layer\_points() %>%

add\_axis("x", title = "Предсказанные значения") %>%

add\_axis("y", title = "Стьюдентизированные остатки")



# Выбросы

Auto %>%

mutate(rstud = ifelse(abs(rstudent(autolm)) > 3, 'Больше 3', 'Не больше 3'),

label = row.names(Auto),

label = ifelse(rstud == 'Больше 3', label, '')) %>%

ggvis(~kpl, ~ horsepower, fill = ~rstud) %>%

layer\_points() %>%

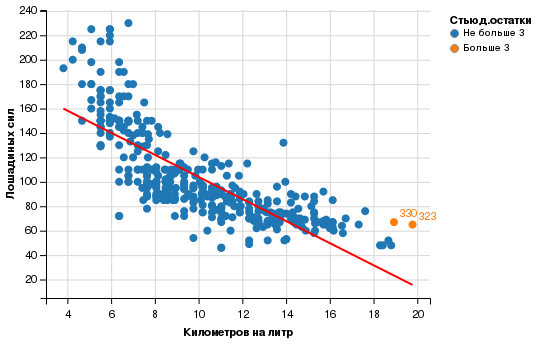
add\_axis("x", title = "Километров на литр") %>%

add\_axis("y", title = "Лошадиных сил") %>%

add\_legend("fill", title = "Стьюд.остатки") %>%

layer\_model\_predictions(stroke := "red", model = 'lm') %>%

layer\_text(text := ~label, dy := -5, dx := 5)



**Точки с избыточным влиянием на модель (High Leverage Points)**

В простой (однофакторной) регрессии мы просто можем посмотреть, какие значения независимой переменной слишком выпадают. Но построим график с оценкой уровня влияния точек.

data.frame(leverage = hatvalues(autolm),

rstuds = rstudent(autolm),

label = row.names(Auto)) %>%

mutate(label = ifelse(leverage > .025, label, '')) %>%

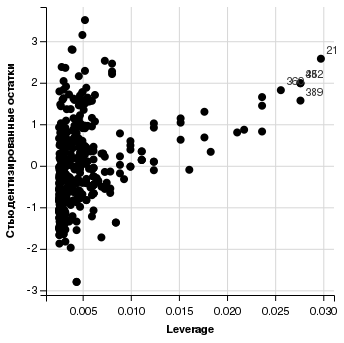
ggvis(~leverage, ~rstuds) %>%

layer\_points() %>%

add\_axis("x", title = "Leverage") %>%

add\_axis("y", title = "Стьюдентизированные остатки") %>%

layer\_text(text := ~label, dy := -5, dx := 5)



# Обогащённая графика с диагностикой модели

data.frame(predict = predict(autolm),

resids = residuals(autolm),

rstuds = rstudent(autolm),

leverage = hatvalues(autolm)) %>%

ggvis(~predict, ~resids, fill = ~abs(rstuds), size = ~leverage) %>%

layer\_points() %>%

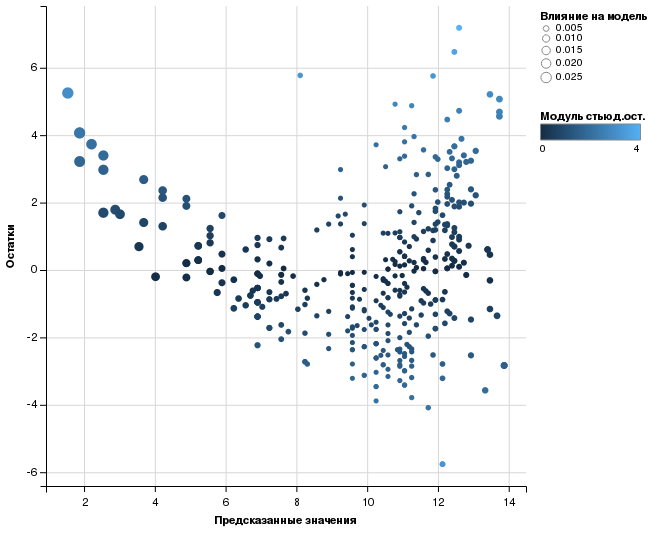
add\_axis("x", title = "Предсказанные значения") %>%

add\_axis("y", title = "Остатки") %>%

add\_legend("size", title = "Влияние на модель") %>%

add\_legend("fill", title = "Модуль стьюд.ост.",

properties = legend\_props(legend = list(y = 100)))



Чем больше точка, тем сильнее её влияние на модель. Чем ярче точка, тем больше у этого наблюдения ошибка предсказания. Закраску точки можно и убрать, поскольку величина остатков коррелирует с величиной стьюдентизированных остатков.

Аналогия в зунек

<http://zoonek2.free.fr/UNIX/48_R/09.html>

варианты решения Хубер, LOESS, ,,,,IRLS, quantile RIDGE

<http://zoonek2.free.fr/UNIX/48_R/10.html>

ПРИМЕРЫ

Выбор моделей регрессии VIF,stepwise пошаговая

<http://r-statistics.co/Model-Selection-in-R.html>

**Квантильная регрессия**

http://bdemeshev.github.io/r\_cycle/cycle\_files/07\_quant\_reg.html

<http://r-statistics.co/adv-regression-models.html>

МУЛЬТИКОЛЛИНЕАРНОСТЬ И LASSO

http://bdemeshev.github.io/r\_cycle/cycle\_files/05\_multicollinearity.html

ГЕТЕРОСКЕДАСТИЧНОСТЬ

http://bdemeshev.github.io/r\_cycle/cycle\_files/12\_hetero.html

# ПРИМЕР QSAR

Самим студентам проделать руками

Пример работы с регрессией, случайный лес для задач классификации

<http://qsar4u.com/files/rintro/03.html>

работа данных

<http://qsar4u.com/files/rintro/02.html>

https://datascienceplus.com/get-your-data-into-r/

# NONLINEAR REGRESSION

<https://datascienceplus.com/second-step-with-non-linear-regression-adding-predictors/>

# Logistic Regression with R: step by step implementation part-1

<https://www.r-bloggers.com/logistic-regression-with-r-step-by-step-implementation-part-1/>

<https://www.r-bloggers.com/how-to-perform-a-logistic-regression-in-r/>

training.data.raw <- read.csv('train.csv',header=T,na.strings=c(""))

sapply(training.data.raw,function(x) sum(is.na(x)))

sapply(training.data.raw, function(x) length(unique(x)))

library(Amelia)

missmap(training.data.raw, main = "Missing values vs observed")

data <- subset(training.data.raw,select=c(2,3,5,6,7,8,10,12))

**Taking care of the missing values**

Now we need to account for the other missing values. R can easily deal with them when fitting a generalized linear model by setting a parameter inside the fitting function. However, personally I prefer to replace the NAs “by hand”, when is possible. There are different ways to do this, a typical approach is to replace the missing values with the average, the median or the mode of the existing one. I’ll be using the average.

data$Age[is.na(data$Age)] <- mean(data$Age,na.rm=T)

As far as categorical variables are concerned, using the read.table() or read.csv() by default will encode the categorical variables as factors. A factor is how R deals categorical variables.  
We can check the encoding using the following lines of code

is.factor(data$Sex)

*TRUE*

is.factor(data$Embarked)

*TRUE*

For a better understanding of how R is going to deal with the categorical variables, we can use the contrasts() function. This function will show us how the variables have been dummyfied by R and how to interpret them in a model.

contrasts(data$Sex)

*male*

*female 0*

*male 1*

contrasts(data$Embarked)

*Q S*

*C 0 0*

*Q 1 0*

*S 0 1*

For instance, you can see that in the variable sex, female will be used as the reference. As for the missing values in Embarked, since there are only two, we will discard those two rows (we could also have replaced the missing values with the mode and keep the datapoints).

data <- data[!is.na(data$Embarked),]

rownames(data) <- NULL

Before proceeding to the fitting process, let me remind you how important is *cleaning and formatting of the data*. This preprocessing step often is crucial for obtaining a good fit of the model and better predictive ability

**Model fitting**

We split the data into two chunks: training and testing set. The training set will be used to fit our model which we will be testing over the testing set.

train <- data[1:800,]

test <- data[801:889,]

Now, let’s fit the model. Be sure to specify the parameter family=binomial in the glm() function.

model <- glm(Survived ~.,family=binomial(link='logit'),data=train)

By using function summary() we obtain the results of our model:

summary(model)

**Interpreting the results of our logistic regression model**

Now we can analyze the fitting and interpret what the model is telling us.  
First of all, we can see that *SibSp*, *Fare* and *Embarked* are not statistically significant. As for the statistically significant variables, sex has the lowest p-value suggesting a strong association of the sex of the passenger with the probability of having survived. The negative coefficient for this predictor suggests that all other variables being equal, the male passenger is less likely to have survived. Remember that in the logit model the response variable is log odds: ln(odds) = ln(p/(1-p)) = a\*x1 + b\*x2 + … + z\*xn. Since male is a dummy variable, being male reduces the log odds by 2.75 while a unit increase in age reduces the log odds by 0.037.

Now we can run the anova() function on the model to analyze the table of deviance

anova(model, test="Chisq")

The difference between the null deviance and the residual deviance shows how our model is doing against the null model (a model with only the intercept). The wider this gap, the better. Analyzing the table we can see the drop in deviance when adding each variable one at a time. Again, adding Pclass, Sex and Age significantly reduces the residual deviance. The other variables seem to improve the model less even though SibSp has a low p-value. A large p-value here indicates that the model without the variable explains more or less the same amount of variation. Ultimately what you would like to see is a significant drop in deviance and the AIC.

While no exact equivalent to the R2 of linear regression exists, the McFadden R2 index can be used to assess the model fit.

library(pscl)

pR2(model)

*Accuracy 0.842696629213483"*

**Assessing the predictive ability of the model**

In the steps above, we briefly evaluated the fitting of the model, now we would like to see how the model is doing when predicting *y* on a new set of data. By setting the parameter type='response', R will output probabilities in the form of P(y=1|X). Our decision boundary will be 0.5. If P(y=1|X) > 0.5 then y = 1 otherwise y=0. Note that for some applications different thresholds could be a better option.

fitted.results <- predict(model,newdata=subset(test,select=c(2,3,4,5,6,7,8)),type='response')

fitted.results <- ifelse(fitted.results > 0.5,1,0)

misClasificError <- mean(fitted.results != test$Survived)

print(paste('Accuracy',1-misClasificError))

The 0.84 accuracy on the test set is quite a good result. However, keep in mind that this result is somewhat dependent on the manual split of the data that I made earlier, therefore if you wish for a more precise score, you would be better off running some kind of cross validation such as k-fold cross validation.

As a last step, we are going to plot the *ROC curve* and calculate the *AUC* (area under the curve) which are typical performance measurements for a binary classifier.  
The ROC is a curve generated by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings while the AUC is the area under the ROC curve. As a rule of thumb, a model with good predictive ability should have an AUC closer to 1 (1 is ideal) than to 0.5.

library(ROCR)

p <- predict(model, newdata=subset(test,select=c(2,3,4,5,6,7,8)), type="response")

pr <- prediction(p, test$Survived)

prf <- performance(pr, measure = "tpr", x.measure = "fpr")

plot(prf)

auc <- performance(pr, measure = "auc")

auc <- auc@y.values[[1]]

auc

*0.8647186*

And here is the ROC plot:

THIS EXAMP

<https://gist.github.com/mick001/ac92e7c017aecff216fd#file-logistic_regression-r>

**ПРИМЕР про женщин у плиты**

<https://bdemeshev.github.io/r_cycle/cycle_files/03_logit_regression.html>

library("mfx") # подсчет предельных эффектов

library("ggplot2") # графики

library("lmtest") # линейные регрессии

library("foreign") # чтение файлов в некоторых форматах

library("dplyr") # манипуляции с таблицами

library("broom") # описание модели в виде таблички

Графики красивые

<https://bdemeshev.github.io/r_cycle/cycle_files/04_qualit_graphs.html>

# КЛАССИФИКАЦИЯ. МЕТОД ОПОРНЫХ ВЕКТОРОВ

**Линейные модели**

[**http://www.algorithmist.ru/2011/07/support-vector-machines-with-examples.html**](http://www.algorithmist.ru/2011/07/support-vector-machines-with-examples.html)

**SVM-pdf в папке**

[**http://datascientist.one/support-vector-machines/**](http://datascientist.one/support-vector-machines/)

**YOUTUBE**

[**https://www.youtube.com/watch?v=HFj128PAZek**](https://www.youtube.com/watch?v=HFj128PAZek)

# пример от демешева

**данные ввод в папке**

[**https://r-analytics.blogspot.com/2011/07/r\_24.html#.W5DmH8J9jIU**](https://r-analytics.blogspot.com/2011/07/r_24.html#.W5DmH8J9jIU)

# ПРИМЕР про квартиры

[**https://bdemeshev.github.io/r\_cycle/cycle\_files/11\_svm.html**](https://bdemeshev.github.io/r_cycle/cycle_files/11_svm.html)

**chem <- read.table(file = "flats\_moscow.txt", header = TRUE)**

**красивые картинки**

**сравнение logit c svm на примере квартир**

[**https://bdemeshev.github.io/r\_cycle/cycle\_files/04\_qualit\_graphs.html**](https://bdemeshev.github.io/r_cycle/cycle_files/04_qualit_graphs.html)

**caret package**

[**http://qsar4u.com/files/rintro/04.html**](http://qsar4u.com/files/rintro/04.html) **(caret)**

**Цикл материалов по логистке, LASSO,**

**Overview**

[**https://freakonometrics.hypotheses.org/52731**](https://freakonometrics.hypotheses.org/52731)

[**https://freakonometrics.hypotheses.org/52747**](https://freakonometrics.hypotheses.org/52747)

**SPLINES REGRESS**

[**https://freakonometrics.hypotheses.org/52771**](https://freakonometrics.hypotheses.org/52771)

[**https://freakonometrics.hypotheses.org/52894**](https://freakonometrics.hypotheses.org/52894)

**SVM**

[**https://freakonometrics.hypotheses.org/52775**](https://freakonometrics.hypotheses.org/52775)

**BAGGING RF**

[**https://freakonometrics.hypotheses.org/52777**](https://freakonometrics.hypotheses.org/52777)

**BOOSTING**

**https://freakonometrics.hypotheses.org/52782**

**TREES**

[**https://freakonometrics.hypotheses.org/52776**](https://freakonometrics.hypotheses.org/52776)

**NN**

[**https://freakonometrics.hypotheses.org/52774**](https://freakonometrics.hypotheses.org/52774)

**Naïve Bayes**

[**https://freakonometrics.hypotheses.org/53021**](https://freakonometrics.hypotheses.org/53021)

**Домашнее задание**

1. Реализовать собственную функцию create.folds для случайного деления выборки на фолды с целью кросс-валидации. Результатом должен быть список, каждый элемент которого содержит индексы соединений, входящих в обучающую выборку отдельного фолда.
   * Реализовать случайную генерацию фолдов с возможность ручной инициализации генератора случайных чисел.
   * Реализовать генерацию фолдов для отсортированных значений. Таким образом, чтобы фолды были сбалансированы с точки зрения распределения активности.
   * Реализовать генерацию фолдов для классификационных задач.
   * Реализовать генерацию сбалансированных фолдов для классификационных задач (чтобы количество представителей разных классов в фолдах было приблизительно одинаковым).
   * Реализовать все эти подходы в виде одной функции create.folds.
2. Осуществить кросс-валидацию для модели randomForest на данных представленных в уроке (df), чтобы сравнить ее с остальными моделями.
3. Использовать файл ENB2012\_data.xlsx в качестве источника данных (для загрузки данных можно использовать пакет xlsx). Построить модели зависимости heating.load и Cooling.load от остальных переменных методами, описанными в уроке (можно также поискать другие методы по приведенной в конце занятия ссылке и использовать их).

library(tidyverse)

df <- data.frame(x=0, y=0)

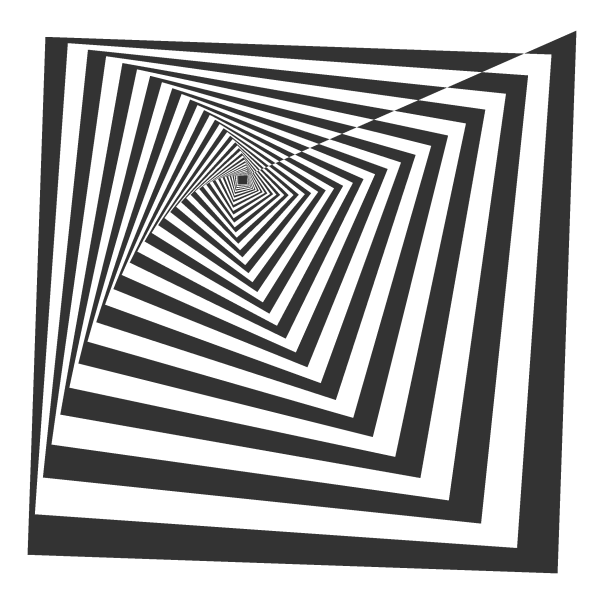
for (i in 2:500){

df[i,1] <- df[i-1,1]+((0.98)^i)\*cos(i)

df[i,2] <- df[i-1,2]+((0.98)^i)\*sin(i)

}

Ggplot(df, aes(x,y)) + geom\_polygon()+theme\_void()



library(tidyverse)

seq(-3,3,by=.01) %>%

  expand.grid(x=., y=.) %>%

  ggplot(aes(x=(x^3-sin(y^2)), y=(y^3-cos(x^2)))) +

  geom\_point(alpha=.1, shape=20, size=0, color="white")+

  theme\_void()+

  coord\_fixed()+

  theme(panel.background = element\_rect(fill="black"))+

  coord\_polar()

library(ggplot2)

a=pi\*(3-sqrt(5))

n=500

ggplot(data.frame(r=sqrt(1:n),t=(1:n)\*a),

       aes(x=r\*cos(t),y=r\*sin(t)))+

  geom\_point(aes(x=0,y=0),

             size=190,

             colour="violetred")+

  geom\_point(aes(size=(n-r)),

             shape=21,fill="gold",

             colour="gray90")+

  theme\_void()+theme(legend.position="none")

library(tidyverse)

seq(-10, 10, by = .05) %>%

  expand.grid(x=., y=.) %>%

  ggplot(aes(x=(x+sin(y)), y=(y+cos(x)))) +

  geom\_point(alpha=.1, shape=20, size=0, color="white")+

  theme\_void()+

  coord\_fixed()+

  theme(panel.background = element\_rect(fill="violetred4")

library(tidyverse)

seq(-10, 10, by = .05) %>%

  expand.grid(x=., y=.) %>%

  ggplot(aes(x=(x+sin(y)), y=(y+cos(x)))) +

  geom\_point(alpha=.1, shape=20, size=0, color="white")+

  theme\_void()+

  coord\_fixed()+

  theme(panel.background = element\_rect(fill="violetred4"))

<https://fronkonstin.com/2016/03/01/a-silky-drawing-and-a-tiny-experiment/>