

# Организационные моменты

# Прокопьева Мария Артуровна

Бакалавриат: Психология (НИУ ВШЭ) '23

Магистратура: Data Science (НИУ ВШЭ) '25

Центр языка и мозга НИУ ВШЭ

tg: [https://t.me/maria\\_protopova](https://t.me/maria_protopova)

mail: [protopovamaria@gmail.com](mailto:protopovamaria@gmail.com)



# План занятий

Дата	Тема	Тип занятия
14.01.2025	Введение в R	Лекция + семинар
	Визуализация данных	Лекция + семинар (дз1)
21.01.2025	Описательная статистика, нормальное распределение, сравнение средних, гипотезы	Лекция + семинар
	Параметрические и непараметрические методы сравнения средних (t-test, ANOVA, M-U, Kruskal_Wallis)	Лекция + семинар (дз2)
28.01.2025	Корреляционный анализ и Линейная регрессия	Лекция
	Корреляционный анализ и Линейная регрессия	Семинар (дз3)
04.02.2025	Множественная линейная регрессия	Лекция
	Множественная линейная регрессия	Семинар (дз4)

# План занятий

Дата	Тема	Тип занятия
11.02.2025	Логистическая регрессия	Лекция
	Логистическая регрессия	Семинар (дз5)
18.02.2025	Линейные модели со смешанными эффектами	Лекция
	Линейные модели со смешанными эффектами	Семинар (дз6)
25.02.2025		Лекция
		Семинар
04.03.2025	Презентация проектов	Семинар
	Презентация проектов	Семинар

# План занятий

- Разбор ДЗ
- Лекция
- Семинар

# Оценивание

- Домашние задания
- Проект
- Бонусы за квизы / участие в экспериментах



```
RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
Flights - RStudio

flights-example.R x
Source on Save Run Source
1 library(nycflights13) ## package containing flights dataset
2 library(lubridate)
3 library(dplyr)
4 library(ggplot2)
5
6 head(flights, n = 3)
7 daily <- flights %>%
8   mutate(date = make_date(year, month, day)) %>%
9   count(date) %>%
10  mutate(wday = wday(date, label = TRUE))
11 head(daily, n = 3)
12 ggplot(daily, aes(wday, n)) +
13   geom_boxplot(outlier.colour = "hotpink") +
14   labs(x = "Weekday", y = "Flights",
15        subtitle = "Number of 2013 New York Flights Each Weekday")
16
1:1 (Top Level) R Script
```

```
Console Terminal x Jobs x
~/Documents/Flights/
# A tibble: 3 x 19
  year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay carrier
  <int> <int> <int>   <int>         <int>      <dbl>   <int>         <int>      <dbl>   <chr>
1  2013     1     1     517           515         2      830           819        11    UA
2  2013     1     1     533           529         4      850           830        20    UA
3  2013     1     1     542           540         2      923           850        33    AA
# ... with 9 more variables: flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
#   distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dtm>
> daily <- flights %>%
+   mutate(date = make_date(year, month, day)) %>%
+   count(date) %>%
+   mutate(wday = wday(date, label = TRUE))
> head(daily, n = 3)
# A tibble: 3 x 3
  date           n wday
  <date>     <int> <ord>
1 2013-01-01   842 Tue
2 2013-01-02   943 Wed
3 2013-01-03   914 Thu
> ggplot(daily, aes(wday, n)) +
+   geom_boxplot(outlier.colour = "hotpink") +
+   labs(x = "Weekday", y = "Flights",
+        subtitle = "Number of 2013 New York Flights Each Weekday")
>
```

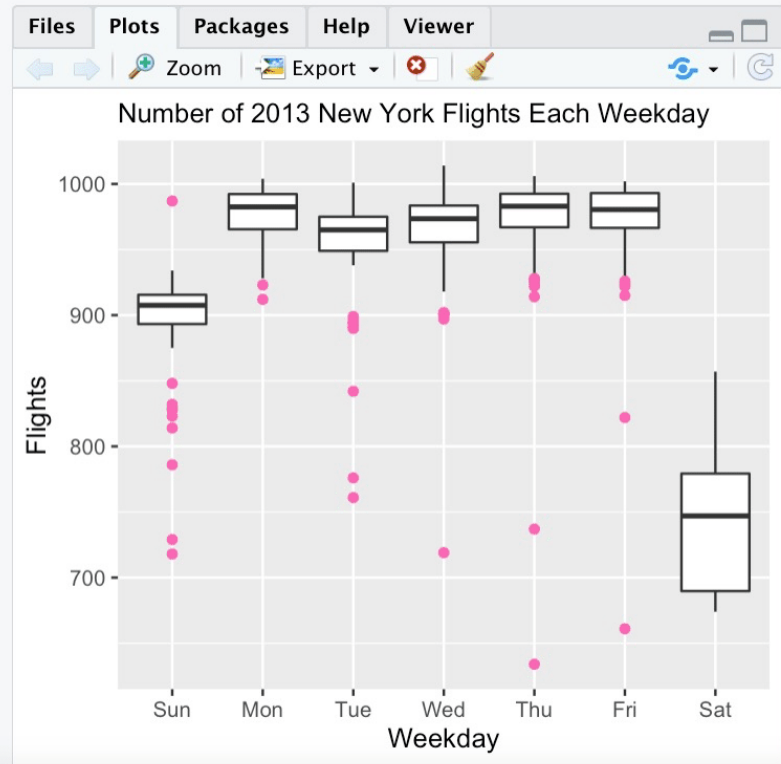
Environment History Connections Tutorial

Global Environment

Data

daily 365 obs. of 3 variables

\$ date:	Date[1:365], format: "2013-01-01" "2013-01-02" ...
\$ n :	int [1:365] 842 943 914 915 720 832 933 899 902...
\$ wday:	Ord.factor w/ 7 levels "Sun"<"Mon"<"Tue"<...: 3 ...





# Basics

## Названия переменных

- 1) Выбирайте краткие и понятные названия переменных
- 2) Название не должно начинаться с числа
- 3) Не используйте в названии пунктуацию, кроме точки (.), тире (-), подчеркивания (\_)
- 4) Не используйте пробелы в названии
- 5) Соблюдайте однородность в коде (*varname*, *VARNAME*, *var.name*, *var\_name*, *VarName*)

# Basics

Символ	Значение
=	Присвоение
<-	Присвоение
==	Равно
!=	Не равно
< (>)	Больше (меньше)
<= (>=)	Больше (меньше) или равно
%in%	Совпадают ли значения? Входит ли значение в набор (список/кортеж) других значений (TRUE / FALSE)?
	Логическое ИЛИ
&	Логическое И

# Errors

- 1) Не бойтесь читать описания ошибок
- 2) Проверьте, загрузили ли вы все библиотеки
- 3) R чувствителен к регистру: *View()* vs. *view()*
- 4) Проверьте раскладку: *c()* vs. *c()*
- 5) Проверьте очепятки
- 6) One line at a time
- 7) Не бойтесь гуглить ошибки
- 8) Не бойтесь задавать «глупые» вопросы и просить о помощи :)



```
> dat <- read.csv("data.csv")
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
  cannot open file 'data.csv': No such file or directory
> |
```



Stack Overflow

3 answers · 11 years ago

**Error: unexpected symbol/input/string constant/numeric ...**

These errors mean that the R code you are trying to run or source is not syntactically correct. That is, you have a typo. [Read more](#)

3 answers · Top answer: These errors mean that the R code you are trying to run or source is not synt...

**Error: unexpected symbol in: R error message - Stack ...** 1 answer May 24, 2018

Wow, a different  
error message...  
Finally some progress!



# Типы данных

# Типы данных

**Тип данных** – характеристика данных, которая определяет множество допустимых значений и набор операций, которые можно с ними осуществлять

# Типы данных

**Номинальные** – данные представляют собой категории, для которых не применимо упорядочивание

НЕ можем ответить на вопрос: Что больше? На сколько больше? Во сколько раз больше?

Например: имя (Ваня, Николь, Пнина), место рождения (Москва, Антананариву, Самарканд), должность (сис.админ, завлаб, водитель)

# Типы данных

**Порядковые** – значения являются дискретными упорядоченными категориями, т.е. порядок категорий имеет смысл. При этом разница между соседними категориями не установлена

Можем ответить на вопрос: Что больше?

НЕ можем ответить на вопрос: На сколько больше? Во сколько раз больше?

Например: уровень образования (среднее, высшее, кандидат наук), шкала Ликерта



# Типы данных

**Интервальные** – числовая шкала, в которой мы можем численно выразить и сравнить разницу между значениями. При этом истинного нуля в этой шкале не существует.

Можем ответить на вопрос: на сколько больше?

Например: градусы Цельсия

# Типы данных

**Шкала отношений** – упорядоченные значения, разница между соседними значениями фиксирована, существует абсолютный ноль.

Можем ответить на вопросы: на сколько больше/меньше? Во сколько больше/меньше?

Например: градусы по Фаренгейту, метр, грамм

Шкала	Описание	Данные
Номинативная	Категории (без порядка)	качественные
Порядковая	Упорядоченные категории	
Интервальная	Можно рассматривать разницу между измерениями, но нет абсолютного нуля	количественные
Шкала отношений	Можно рассматривать разницу между измерениями, есть истинный ноль	

# Типы данных в R

- numeric: 1, 3.14, -0.777778
- logical: TRUE, FALSE
- character: “*i <3 R programming language*”
- factor – это строковые данные, которые хранятся в виде чисел, так как содержат небольшое количество уникальных значений → используются для задания групп в данных

# Структуры данных

Структура	R	Пример
вектор	c()	c(1,2,3)
список	list()	list(1, TRUE, “stuff”)
матрица	matrix()	matrix(c(1,2,3,4), ncol=2) matrix(c(1,2,3,4), nrow=1)
датафрейм	data.frame()	data.frame(name=c(“Маша”, “Bob”), age=c(25, 28), city=c(”Москва”, “Springfield”))
	tibble::tibble()	tibble(age = sample(18:60, size = 30), group = ifelse(age > 40, 'old', 'young'))

# Предобработка данных

# Импорт данных

```
read_csv("path/to/file") # для чтения классических csv-файлов  
read_csv2("path/to/file") # для чтения csv-файлов с разделителем точка в запятой  
read_delim("path/to/file", delim = "...") # для чтения файлов с любым разделителем
```

```
raw01 <- readxl::read_xlsx('data/data_sharexp/01.xlsx', 2)
```

# Смотрим на данные

- **View()**
- **str()**
- **head()**
- **tail()**
- **nrow()**
- **ncol()**



# library(tidyverse)

- Pipe (%>%) – оператор, который передает в качестве первого аргумента то, что стоит слева, в функцию, которая стоит справа

```
> sum(5, 6)
[1] 11
> 5 %>% sum(6)
[1] 11

> sqrt(abs(log(abs(round(sin(1 / cos(3)), 2)), 3)))
[1] 0.3846181
> 3 %>% cos() %>%
+   `/`(1, .) %>%
+   sin() %>%
+   round(2) %>%
+   abs() %>%
+   log(3) %>%
+   abs() %>%
+   sqrt()
[1] 0.3846181

data[data$train_test=='TEST',] %>%
  group_by(
    latency_class,
    duration_class,
    subject,
    N_repetition,
    N_training,
    decimation
  ) %>%
  summarize(
    percent_corr = mean(percent_corr)
  ) %>%
  ungroup() -> data_mean
```

# library(tidyverse)

- **select()**
- **rename()**
- **sapply()**
- **lapply()**
- **filter()**
- **mutate()**

# Соединение датасетов

- **left\_join()** – левый датасет остается полностью, к нему присоединяются строки правого датасета, соответствующие условию
- **right\_join()** – правый датасет остается полностью, к нему присоединяются строки левого датасета, соответствующие условию
- **full\_join()** – остаются все строки обоих датасетов, а те, которым не нашлось соответствия в одном из датасетов, ставится NA
- **inner\_join()** – остаются только те строки обоих датасетов, которым нашлось соответствие
- **anti\_join()** – остаются только те строки обоих датасетов, которым не нашлось соответствие

# Группировка и агрегация данных

Агрегация данных – объединение данных по группам и подсчет каких-то показателей внутри каждой группы

- **group\_by()**
- **ungroup()**
- **summarize()**

# Широкий и длинный форматы

Чаще всего, мы сталкиваемся с данными в таком виде, что по строкам идут наблюдения, а по столбцам – переменные. Такой формат данных называется *широким* (потому что переменных может быть много). В *длинном* формате есть столбец идентификатора, столбец переменных и столбец значений этих переменных

- **pivot\_longer()**
- **pivot\_wider()**

Wide Data Format

Groups	Lab	Theory
A	25	60
B	21	55
C	14	49
D	22	69

Every value is unique in first column

The values in First column repeat

Long Data Format

Groups	Variable	Values
A	Lab	25
B	Lab	21
C	Lab	14
D	Lab	22
A	Theory	60
B	Theory	55
C	Theory	49
D	Theory	69

# Практика на занятии

Using the diamonds built-in dataset (requires **tidyverse**), perform the following tasks:

- 1) View all the variable names in diamonds.
- 2) Create a new variable named salePrice to reflect a discount of \$250 off of the original cost of each diamond
- 3) Remove the x, y, and z variables from the diamonds dataset
- 4) Determine the number of diamonds there are for each cut value
- 5) Create a new column named totalNum that calculates the total number of diamonds.
- 6) Create a new variable called Acceptable with yes or no values using ifelse(). For a yes value, the following criteria must be met:
  - Carat  $\leq 0.8$  and Depth  $> 62$
  - Carat  $> 1$