Introduction to R, Season 2

January, 2024

Table of Contents

# 1 About this Course

## 1.1 Curriculum

The course covers fundamentals of R, a high-level programming language, and use it to wrangle data for analysis and visualization.

## 1.2 Target Audience

The course is intended for researchers who want to learn coding for the first time with a data science application, or have explored programming and want to focus on fundamentals.

# 2 Intro to Computing

## 2.1 Goals of the course

* Fundamental concepts in high-level programming languages (R, Python, Julia, WDL, etc.) that is transferable: *How do programs run, and how do we solve problems using functions and data structures?*
* Beginning of data science fundamentals: *How do you translate your scientific question to a data wrangling problem and answer it?*
* 
* Figure : Data science workflow
* Find a nice balance between the two throughout the course: we will try to reproduce a figure from a scientific publication using new data.

## 2.2 What is a computer program?

* A sequence of instructions to manipulate data for the computer to execute.
* A series of translations: English <-> Programming Code for Interpreter <-> Machine Code for Central Processing Unit (CPU)

We will focus on English <-> Programming Code for R Interpreter in this class.

More importantly: **How we organize ideas <-> Instructing a computer to do something**.

## 2.3 A programming language has following elements:

* Grammar structure to construct expressions
* Combining expressions to create more complex expressions
* Encapsulate complex expressions via functions to create modular and reusable tasks
* Encapsulate complex data via data structures to allow efficient manipulation of data

## 2.4 Posit Cloud Setup

Posit Cloud/RStudio is an Integrated Development Environment (IDE). Think about it as Microsoft Word to a plain text editor. It provides extra bells and whistles to using R that is easier for the user.

Today, we will pay close attention to:

* Script editor: where sequence of instructions are typed and saved as a text document as a R program. To run the program, the console will execute every single line of code in the document.
* Console (interpreter): Instead of giving a entire program in a text file, you could interact with the R Console line by line. You give it one line of instruction, and the console executes that single line. It is what R looks like without RStudio.
* Environment: Often, code will store information *in memory*, and it is shown in the environment. More on this later.

## 2.5 Using Quarto for your work

Why should we use Quarto for data science work?

* Encourages reproducible workflows
* Code, output from code, and prose combined together
* Extensions to Python, Julia, and more.

More options and guides can be found in [Introduction to Quarto](https://quarto.org/docs/get-started/hello/rstudio.html) .

## 2.6 Grammar Structure 1: Evaluation of Expressions

* **Expressions** are be built out of **operations** or **functions**.
* Operations and functions combine **data types** to return another data type.
* We can combine multiple expressions together to form more complex expressions: an expression can have other expressions nested inside it.

For instance, consider the following expressions entered to the R Console:

18 + 21

## [1] 39

max(18, 21)

## [1] 21

max(18 + 21, 65)

## [1] 65

18 + (21 + 65)

## [1] 104

nchar("ATCG")

## [1] 4

Here, our input **data types** to the operation are **numeric** in lines 1-4 and our input data type to the function is **character** in line 5.

Operations are just functions in hiding. We could have written:

sum(18, 21)

## [1] 39

sum(18, sum(21, 65))

## [1] 104

Remember the function machine from algebra class? We will use this schema to think about expressions.



Figure : Function machine from algebra class.

If an expression is made out of multiple, nested operations, what is the proper way of the R Console interpreting it? Being able to read nested operations and nested functions as a programmer is very important.

3 \* 4 + 2

## [1] 14

3 \* (4 + 2)

## [1] 18

Lastly, a note on the use of functions: a programmer should not need to know how the function is implemented in order to use it - this emphasizes [abstraction and modular thinking](#X4289ef2db5109e8791918a3c7b0cce6f2f89fd2), a foundation in any programming language.

### 2.6.1 Data types

Here are some data types that we will be using in this course:

* **Numeric**: 18, 21, 65, 1.25
* **Character**: “ATCG”, “Whatever”, “948-293-0000”
* **Logical**: TRUE, FALSE

## 2.7 Grammar Structure 2: Storing data types in the environment

To build up a computer program, we need to store our returned data type from our expression somewhere for downstream use. We can assign a variable to it as follows:

x = 18 + 21

If you enter this in the Console, you will see that in the Environment, the variable x has a value of 39.

### 2.7.1 Execution rule for variable assignment

Evaluate the expression to the right of =.

Bind variable to the left of = to the resulting value.

The variable is stored in the environment.

<- is okay too!

The environment is where all the variables are stored, and can be used for an expression anytime once it is defined. Only one unique variable name can be defined.

The variable is stored in the working memory of your computer, Random Access Memory (RAM). This is temporary memory storage on the computer that can be accessed quickly. Typically a personal computer has 8, 16, 32 Gigabytes of RAM. When we work with large datasets, if you assign a variable to a data type larger than the available RAM, it will not work. More on this later.

Look, now x can be reused downstream:

x - 2

## [1] 37

y = x \* 2

## 2.8 Grammar Structure 3: Evaluation of Functions

A function has a **function name**, **arguments**, and **returns** a data type.

### 2.8.1 Execution rule for functions:

Evaluate the function by its arguments, and if the arguments are functions or contains operations, evaluate those functions or operations first.

The output of functions is called the **returned value**.

sqrt(nchar("hello"))

## [1] 2.236068

(nchar("hello") + 4) \* 2

## [1] 18

## 2.9 Tips on writing your first code

Computer = powerful + stupid

Even the smallest spelling and formatting changes will cause unexpected output and errors!

* Write incrementally, test often
* Check your assumptions, especially using new functions, operations, and new data types.
* Live environments are great for testing, but not great for reproducibility.
* Ask for help!

# 3 Working with data structures

## 3.1 Vectors

In the first exercise, you started to explore **data structures**, which store information about data types. You played around with **vectors**, which is a ordered collection of a data type. Each *element* of a vector contains a data type, and there is no limit on how big a vector can be, as long the memory use of it is within the computer’s memory (RAM).

We can now store a vast amount of information in a vector, and assign it to a single variable. We can now use operations and functions on a vector, modifying many elements within the vector at once! This fits with the feature of “encapsulate complex data via data structures to allow efficient manipulation of data” described in the first lesson!

We often create vectors using the combine function, c() :

staff = c("chris", "shasta", "jeff")  
chrNum = c(2, 3, 1)

If we try to create a vector with mixed data types, R will try to make them be the same data type, or give an error:

staff = c("chris", "shasta", 123)  
staff

## [1] "chris" "shasta" "123"

Our numeric got converted to character so that the entire vector is all characters.

### 3.1.1 Using operations on vectors

Recall from the first class:

* Expressions are be built out of **operations** or **functions**.
* Operations and functions combine **data types** to return another data type.

Now that we are working with data structures, the same principle applies:

* Operations and functions combine **data structures** to return another data structure (or data type!).

What happens if we use some familiar operations we used for numerics on a numerical vector? If we multiply a numerical vector by a numeric, what do we get?

chrNum = chrNum \* 3  
chrNum

## [1] 6 9 3

All of chrNum’s elements tripled! Our multiplication operation, when used on a *numeric vector with a numeric*, has a *new* meaning: it multiplied all the elements by 3. Multiplication is an operation that can be used for multiple data types or data structures: we call this property **operator overloading**. Here’s another example: *numeric vector multiplied by another numeric vector*:

chrNum \* c(2, 2, 0)

## [1] 12 18 0

but there are also limits: a numeric vector added to a character vector creates an error:

#chrNum + staff

When we work with operations and functions, we must be mindful what inputs the operation or function takes in, and what outputs it gives, no matter how “intuitive” the operation or function name is.

### 3.1.2 Subsetting vectors explicitly

In the exercise this past week, you looked at a new operation to subset elements of a vector using brackets.

Inside the bracket is either a single numeric value or an a **numerical indexing vector** containing numerical values. They dictate which elements of the vector to return.

staff[2]

## [1] "shasta"

staff[c(1, 2)]

## [1] "chris" "shasta"

small\_staff = staff[c(1, 2)]

In the last line, we created a new vector small\_staff that is a subset of the staff given the indexing vector c(1, 2). We have three vectors referenced in one line of code. This is tricky and we need to always refer to our rules step-by-step: evaluate the expression right of the =, which contains a vector bracket. Follow the rule of the vector bracket. Then store the returning value to the variable left of =.

Alternatively, instead of using numerical indexing vectors, we can use a **logical indexing vector**. The logical indexing vector must be the *same length* as the vector to be subsetted, with TRUE indicating an element to keep, and FALSE indicating an element to drop. The following block of code gives the same value as before:

staff[c(TRUE, FALSE, FALSE)]

## [1] "chris"

staff[c(TRUE, TRUE, FALSE)]

## [1] "chris" "shasta"

small\_staff = staff[c(TRUE, TRUE, FALSE)]

### 3.1.3 Subsetting vectors implicitly

Here are two applications of subsetting on vectors that need distinction to write the correct code:

1. **Explicit subsetting**: Suppose someone approaches you a 100-length vector of people’s ages, and say that they want to subset to the first 10 elements.
2. **Implicit subsetting**: Suppose someone approaches you a 100-length vector of people’s ages, and say that they want to subset to elements < 18 age.

We already know how to explicitly subset:

set.seed(123) #don't worry about this function  
age = round(runif(100, 1, 100)) #don't worry about these functions  
first\_ten\_age = age[1:10]

For implicit subsetting, we don’t know which elements to select off the top of our head! If we know which elements have less than 18, then we can give the elements for an explicit subset. Therefore, we need to create a logical indexing vector using a **comparison operator**:

indexing\_vector = age < 18  
indexing\_vector

## [1] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  
## [13] FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE  
## [37] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE TRUE FALSE FALSE  
## [49] FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE  
## [61] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [73] FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [85] TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [97] FALSE TRUE FALSE FALSE

The comparison operator < compared the numeric value of age to see which elements of age is less than 18, and then returned a logical vector that has TRUE if age is less than 18 at that element and FALSE otherwise.

Then,

age\_young = age[indexing\_vector]  
age\_young

## [1] 6 11 5 16 3 15 16 15 6 13 14 10 1 12 11 14 10

We could have done this all in one line without storing the indexing vector as a variable in the environment:

age\_young = age[age < 18]

We have the following comparison operators in R:

< less than

<= less or equal than

== equal to

!= not equal to

> greater than

>= greater than or equal to

You can also put these comparison operators together to form more complex statements, which you will explore in this week’s exercise.

Another example:

age\_90 = age[age == 90]  
age\_90

## [1] 90 90 90

age\_not\_90 = age[age != 90]  
age\_not\_90

## [1] 29 79 41 88 94 6 53 89 56 46 96 46 68 58 11 25 5 33 95 89 70 64 99 66 71  
## [26] 55 60 30 16 96 69 80 3 48 76 22 32 24 15 42 42 38 16 15 24 47 27 86 6 45  
## [51] 80 13 57 21 14 76 38 67 10 39 28 82 45 81 81 80 45 76 63 71 1 48 23 39 62  
## [76] 36 12 25 67 42 79 11 44 99 89 89 18 14 66 35 66 33 20 78 10 47 52

For most of our subsetting tasks on vectors (and dataframes below), we will be encouraging implicit subsetting. The power of implicit subsetting is that you don’t need to know what your vector contains to do something with it! This technique is related to *abstraction* in programming mentioned in the first lesson: by using expressions to find the specific value you are interested instead of *hard-coding* the value explicitly, it generalizes your code to handle a wider variety of situations.

## 3.2 Dataframes

Before we dive into dataframes, check that the tidyverse package is properly installed by loading it in your R Console:

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.3

## Warning: package 'purrr' was built under R version 4.0.5

## Warning: package 'stringr' was built under R version 4.0.3

Here is the data structure you have been waiting for: the **dataframe**. A dataframe is a spreadsheet such that each column must have the same data type. Think of a bunch of vectors organized as columns, and you get a dataframe.

For the most part, we load in dataframes from a file path (although they are sometimes created by combining several vectors of the same length, but we won’t be covering that here):

load(url("https://github.com/fhdsl/S1\_Intro\_to\_R/raw/main/classroom\_data/CCLE.RData"))

### 3.2.1 Using functions and operations on dataframes

We can run some useful functions on dataframes to get some useful properties, similar to how we used length() for vectors:

nrow(metadata)

## [1] 1864

ncol(metadata)

## [1] 30

dim(metadata)

## [1] 1864 30

colnames(metadata)

## [1] "ModelID" "PatientID" "CellLineName"   
## [4] "StrippedCellLineName" "Age" "SourceType"   
## [7] "SangerModelID" "RRID" "DepmapModelType"   
## [10] "AgeCategory" "GrowthPattern" "LegacyMolecularSubtype"  
## [13] "PrimaryOrMetastasis" "SampleCollectionSite" "Sex"   
## [16] "SourceDetail" "LegacySubSubtype" "CatalogNumber"   
## [19] "CCLEName" "COSMICID" "PublicComments"   
## [22] "WTSIMasterCellID" "EngineeredModel" "TreatmentStatus"   
## [25] "OnboardedMedia" "PlateCoating" "OncotreeCode"   
## [28] "OncotreeSubtype" "OncotreePrimaryDisease" "OncotreeLineage"

The last function, colnames() returns a character vector of the column names of the dataframe. This is an important property of dataframes that we will make use of to subset on it.

We introduce an operation for dataframes: the dataframe$column\_name operation selects for a column by its column name and returns the column as a vector. For instance:

metadata$OncotreeLineage[1:5]

## [1] "Ovary/Fallopian Tube" "Myeloid" "Bowel"   
## [4] "Myeloid" "Myeloid"

metadata$Age[1:5]

## [1] 60 36 72 30 30

We treat the resulting value as a vector, so we can perform implicit subsetting:

metadata$OncotreeLineage[metadata$OncotreeLineage == "Myeloid"]

## [1] "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid"  
## [8] "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid"  
## [15] "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid"  
## [22] "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid"  
## [29] "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid"  
## [36] "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid"  
## [43] "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid"  
## [50] "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid"  
## [57] "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid"  
## [64] "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid"  
## [71] "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid" "Myeloid"

The bracket operation [ ] on a dataframe can also be used for subsetting. dataframe[row\_idx, col\_idx] subsets the dataframe by a row indexing vector row\_idx, and a column indexing vector col\_idx.

metadata[1:5, c(1, 3)]

## ModelID CellLineName  
## 1 ACH-000001 NIH:OVCAR-3  
## 2 ACH-000002 HL-60  
## 3 ACH-000003 CACO2  
## 4 ACH-000004 HEL  
## 5 ACH-000005 HEL 92.1.7

We can refer to the column names directly:

metadata[1:5, c("ModelID", "CellLineName")]

## ModelID CellLineName  
## 1 ACH-000001 NIH:OVCAR-3  
## 2 ACH-000002 HL-60  
## 3 ACH-000003 CACO2  
## 4 ACH-000004 HEL  
## 5 ACH-000005 HEL 92.1.7

We can leave the column index or row index empty to just subset columns or rows.

metadata[1:5, ]

## ModelID PatientID CellLineName StrippedCellLineName Age SourceType  
## 1 ACH-000001 PT-gj46wT NIH:OVCAR-3 NIHOVCAR3 60 Commercial  
## 2 ACH-000002 PT-5qa3uk HL-60 HL60 36 Commercial  
## 3 ACH-000003 PT-puKIyc CACO2 CACO2 72 Commercial  
## 4 ACH-000004 PT-q4K2cp HEL HEL 30 Commercial  
## 5 ACH-000005 PT-q4K2cp HEL 92.1.7 HEL9217 30 Commercial  
## SangerModelID RRID DepmapModelType AgeCategory GrowthPattern  
## 1 SIDM00105 CVCL\_0465 HGSOC Adult Adherent  
## 2 SIDM00829 CVCL\_0002 AML Adult Suspension  
## 3 SIDM00891 CVCL\_0025 COAD Adult Adherent  
## 4 SIDM00594 CVCL\_0001 AML Adult Suspension  
## 5 SIDM00593 CVCL\_2481 AML Adult Mixed  
## LegacyMolecularSubtype PrimaryOrMetastasis SampleCollectionSite  
## 1 Metastatic ascites  
## 2 Primary haematopoietic\_and\_lymphoid\_tissue  
## 3 Primary Colon  
## 4 Primary haematopoietic\_and\_lymphoid\_tissue  
## 5 bone\_marrow  
## Sex SourceDetail LegacySubSubtype CatalogNumber  
## 1 Female ATCC high\_grade\_serous HTB-71  
## 2 Female ATCC M3 CCL-240  
## 3 Male ATCC HTB-37  
## 4 Male DSMZ M6 ACC 11  
## 5 Male ATCC M6 HEL9217  
## CCLEName COSMICID PublicComments  
## 1 NIHOVCAR3\_OVARY 905933   
## 2 HL60\_HAEMATOPOIETIC\_AND\_LYMPHOID\_TISSUE 905938   
## 3 CACO2\_LARGE\_INTESTINE NA   
## 4 HEL\_HAEMATOPOIETIC\_AND\_LYMPHOID\_TISSUE 907053   
## 5 HEL9217\_HAEMATOPOIETIC\_AND\_LYMPHOID\_TISSUE NA   
## WTSIMasterCellID EngineeredModel TreatmentStatus OnboardedMedia PlateCoating  
## 1 2201 MF-001-041 None  
## 2 55 MF-005-001 None  
## 3 NA Unknown MF-015-009 None  
## 4 783 Post-treatment MF-001-001 None  
## 5 NA MF-001-001 None  
## OncotreeCode OncotreeSubtype OncotreePrimaryDisease  
## 1 HGSOC High-Grade Serous Ovarian Cancer Ovarian Epithelial Tumor  
## 2 AML Acute Myeloid Leukemia Acute Myeloid Leukemia  
## 3 COAD Colon Adenocarcinoma Colorectal Adenocarcinoma  
## 4 AML Acute Myeloid Leukemia Acute Myeloid Leukemia  
## 5 AML Acute Myeloid Leukemia Acute Myeloid Leukemia  
## OncotreeLineage  
## 1 Ovary/Fallopian Tube  
## 2 Myeloid  
## 3 Bowel  
## 4 Myeloid  
## 5 Myeloid

metadata[, c("ModelID", "CellLineName")]

## ModelID CellLineName  
## 1 ACH-000001 NIH:OVCAR-3  
## 2 ACH-000002 HL-60  
## 3 ACH-000003 CACO2  
## 4 ACH-000004 HEL  
## 5 ACH-000005 HEL 92.1.7  
## 6 ACH-000006 MONO-MAC-6  
## 7 ACH-000007 LS513  
## 8 ACH-000008 A101D  
## 9 ACH-000009 C2BBe1  
## 10 ACH-000011 253J  
## 11 ACH-000012 HCC-827  
## 12 ACH-000013 ONCO-DG-1  
## 13 ACH-000014 Hs 294T  
## 14 ACH-000015 NCI-H1581  
## 15 ACH-000016 SLR 21  
## 16 ACH-000017 SK-BR-3  
## 17 ACH-000018 T24  
## 18 ACH-000019 MCF7  
## 19 ACH-000020 MHH-CALL-2  
## 20 ACH-000021 NCI-H1693  
## 21 ACH-000022 PA-TU-8988S  
## 22 ACH-000023 PA-TU-8988T  
## 23 ACH-000024 OPM-2  
## 24 ACH-000025 CH-157MN  
## 25 ACH-000026 253J-BV  
## 26 ACH-000027 GOS-3  
## 27 ACH-000028 KPL-1  
## 28 ACH-000029 HCC-827-GR5  
## 29 ACH-000030 PC-14  
## 30 ACH-000031 Panc 02.13  
## 31 ACH-000032 MHH-CALL-3  
## 32 ACH-000033 NCI-H1819  
## 33 ACH-000034 PLB-985  
## 34 ACH-000035 NCI-H1650  
## 35 ACH-000036 U343  
## 36 ACH-000037 S-117  
## 37 ACH-000038 EHEB  
## 38 ACH-000039 SK-N-MC  
## 39 ACH-000040 U-118 MG  
## 40 ACH-000041 RD-ES  
## 41 ACH-000042 Panc 02.03  
## 42 ACH-000043 Hs 895.T  
## 43 ACH-000044 MDA-MB-134-VI  
## 44 ACH-000045 MV4-11  
## 45 ACH-000046 ACHN  
## 46 ACH-000047 GCIY  
## 47 ACH-000048 TOV-112D  
## 48 ACH-000049 HEK TE  
## 49 ACH-000050 NCI-H929  
## 50 ACH-000051 TE 617.T  
## 51 ACH-000052 A-673  
## 52 ACH-000053 KARPAS-299  
## 53 ACH-000054 HT-1080  
## 54 ACH-000055 D283 Med  
## 55 ACH-000056 DOHH-2  
## 56 ACH-000057 OPM-1  
## 57 ACH-000058 ML-1  
## 58 ACH-000059 SUP-B15  
## 59 ACH-000060 Panc 10.05  
## 60 ACH-000061 HH  
## 61 ACH-000062 RERF-LC-MS  
## 62 ACH-000063 Hs 616.T  
## 63 ACH-000064 SALE  
## 64 ACH-000065 OCI-AML5  
## 65 ACH-000066 HCC4006  
## 66 ACH-000067 Hs 683  
## 67 ACH-000068 REC-1  
## 68 ACH-000069 Hs 611.T  
## 69 ACH-000070 697  
## 70 ACH-000071 Hs 706.T  
## 71 ACH-000072 MEG-01  
## 72 ACH-000073 GRANTA-519  
## 73 ACH-000074 KU812  
## 74 ACH-000075 U-87 MG  
## 75 ACH-000076 NCO2  
## 76 ACH-000077 MJ  
## 77 ACH-000078 MHH-NB-11  
## 78 ACH-000079 TE 125.T  
## 79 ACH-000080 BDCM  
## 80 ACH-000081 GDM-1  
## 81 ACH-000082 G-292, clone A141B1  
## 82 ACH-000083 Hs 281.T  
## 83 ACH-000084 MUTZ-3  
## 84 ACH-000085 T3M-4  
## 85 ACH-000086 ACC-MESO-1  
## 86 ACH-000087 SK-ES-1  
## 87 ACH-000088 Hs 172.T  
## 88 ACH-000089 NCI-H684  
## 89 ACH-000090 PC-3  
## 90 ACH-000091 OV56  
## 91 ACH-000092 NCI-H2452  
## 92 ACH-000093 Panc 05.04  
## 93 ACH-000094 HPAF-II  
## 94 ACH-000095 D341 Med  
## 95 ACH-000096 G-401  
## 96 ACH-000097 ZR-75-1  
## 97 ACH-000098 GAMG  
## 98 ACH-000099 SIMA  
## 99 ACH-000100 RH-41  
## 100 ACH-000101 KE-37  
## 101 ACH-000102 GMS-10  
## 102 ACH-000103 Caov-4  
## 103 ACH-000104 Loucy  
## 104 ACH-000105 ALL-SIL  
## 105 ACH-000106 JVM-2  
## 106 ACH-000107 Capan-2  
## 107 ACH-000108 KP-3  
## 108 ACH-000109 NCI-H3255  
## 109 ACH-000110 NCC-StC-K140  
## 110 ACH-000111 HCC1187  
## 111 ACH-000112 SIG-M5  
## 112 ACH-000113 OCI-AML2  
## 113 ACH-000114 SU.86.86  
## 114 ACH-000115 VCaP  
## 115 ACH-000116 OAW28  
## 116 ACH-000117 EFM-192A  
## 117 ACH-000118 HUP-T3  
## 118 ACH-000119 Hs 863.T  
## 119 ACH-000120 CHP-212  
## 120 ACH-000121 NCI-H2405  
## 121 ACH-000122 SUP-T11  
## 122 ACH-000123 COV434  
## 123 ACH-000124 OCI-LY-19  
## 124 ACH-000125 TO 175.T  
## 125 ACH-000126 KG-1-C  
## 126 ACH-000127 SLR 20  
## 127 ACH-000128 LN-319  
## 128 ACH-000129 NCI-H1341  
## 129 ACH-000130 NALM-19  
## 130 ACH-000131 Hs 229.T  
## 131 ACH-000132 JHOS-2  
## 132 ACH-000133 Hs 729  
## 133 ACH-000134 Hs 274.T  
## 134 ACH-000135 Hs 940.T  
## 135 ACH-000136 CHP-126  
## 136 ACH-000137 8-MG-BA  
## 137 ACH-000138 CFPAC-1  
## 138 ACH-000139 Panc 03.27  
## 139 ACH-000140 Pfeiffer  
## 140 ACH-000141 SNU-308  
## 141 ACH-000142 CAL-29  
## 142 ACH-000143 HCC2429  
## 143 ACH-000144 RERF-GC-1B  
## 144 ACH-000145 SK-LMS-1  
## 145 ACH-000146 THP-1  
## 146 ACH-000147 T-47D  
## 147 ACH-000148 Hs 578T  
## 148 ACH-000149 SK-N-SH  
## 149 ACH-000150 HCC2935  
## 150 ACH-000151 JM1  
## 151 ACH-000152 M059K  
## 152 ACH-000153 NCI-H2052  
## 153 ACH-000154 Hs 888.T  
## 154 ACH-000155 SW 1990  
## 155 ACH-000156 MHH-CALL-4  
## 156 ACH-000157 A4/Fuk  
## 157 ACH-000158 OCI-LY3  
## 158 ACH-000159 OS-RC-2  
## 159 ACH-000160 BT-12  
## 160 ACH-000161 COR-L105  
## 161 ACH-000162 GA-10  
## 162 ACH-000163 SW579  
## 163 ACH-000164 PANC-1  
## 164 ACH-000165 Hs 751.T  
## 165 ACH-000166 Kasumi-6  
## 166 ACH-000167 KE-97  
## 167 ACH-000168 NOMO-1  
## 168 ACH-000169 RD  
## 169 ACH-000170 PrEC LH  
## 170 ACH-000171 VMRC-RCZ  
## 171 ACH-000172 TM-87  
## 172 ACH-000173 JHUEM-3  
## 173 ACH-000174 CAL-62  
## 174 ACH-000175 TE159T  
## 175 ACH-000176 LOU-NH91  
## 176 ACH-000177 NCI-H660  
## 177 ACH-000178 Hs 766T  
## 178 ACH-000179 NCI-H1618  
## 179 ACH-000180 Hs 839.T  
## 180 ACH-000181 SCC-9  
## 181 ACH-000182 SNU-869  
## 182 ACH-000183 L-363  
## 183 ACH-000184 Hs 343.T  
## 184 ACH-000185 Hs 737.T  
## 185 ACH-000186 NCI-H2444  
## 186 ACH-000187 COR-L311  
## 187 ACH-000188 SCC-25  
## 188 ACH-000189 RCC10RGB  
## 189 ACH-000190 HD-MY-Z  
## 190 ACH-000191 BHT-101  
## 191 ACH-000192 MFE-280  
## 192 ACH-000193 KARPAS-620  
## 193 ACH-000194 Hs 934.T  
## 194 ACH-000195 Set-2  
## 195 ACH-000196 HCC1599  
## 196 ACH-000197 TALL-1  
## 197 ACH-000198 EOL-1  
## 198 ACH-000199 Hs 255.T  
## 199 ACH-000200 NMC-G1  
## 200 ACH-000201 A-204  
## 201 ACH-000202 COLO-320  
## 202 ACH-000203 NH-6  
## 203 ACH-000204 LP-1  
## 204 ACH-000205 PK-59  
## 205 ACH-000206 C8166  
## 206 ACH-000207 Detroit 562  
## 207 ACH-000208 U-178  
## 208 ACH-000209 SNU-1079  
## 209 ACH-000210 CADO-ES1  
## 210 ACH-000211 Daoy  
## 211 ACH-000212 CAL-120  
## 212 ACH-000213 HUP-T4  
## 213 ACH-000214 Hs 675.T  
## 214 ACH-000215 LN382  
## 215 ACH-000216 JH-EsoAd1  
## 216 ACH-000217 JHH-6  
## 217 ACH-000218 PL-21  
## 218 ACH-000219 A-375  
## 219 ACH-000220 Mino  
## 220 ACH-000221 SNU-398  
## 221 ACH-000222 AsPC-1  
## 222 ACH-000223 HCC1937  
## 223 ACH-000224 Hs 819.T  
## 224 ACH-000225 ECC12  
## 225 ACH-000226 SUP-M2  
## 226 ACH-000227 KP-N-YN  
## 227 ACH-000228 BICR 31  
## 228 ACH-000229 Hs 822.T  
## 229 ACH-000230 Hs 742.T  
## 230 ACH-000231 KALS-1  
## 231 ACH-000232 U-251 MG  
## 232 ACH-000233 DEL  
## 233 ACH-000234 Caki-2  
## 234 ACH-000235 Panc 04.03  
## 235 ACH-000236 SW1417  
## 236 ACH-000237 JHOM-1  
## 237 ACH-000238 SCC-4  
## 238 ACH-000239 HuG1-N  
## 239 ACH-000240 Hs 600.T  
## 240 ACH-000241 JK-1  
## 241 ACH-000242 RT4  
## 242 ACH-000243 DAN-G  
## 243 ACH-000244 DK-MG  
## 244 ACH-000245 BL-41  
## 245 ACH-000246 SLR 23  
## 246 ACH-000247 OCUM-1  
## 247 ACH-000248 AU565  
## 248 ACH-000249 CL-11  
## 249 ACH-000250 KMRC-20  
## 250 ACH-000251 NCI-H2887  
## 251 ACH-000252 LS1034  
## 252 ACH-000253 COLO 201  
## 253 ACH-000254 SCC-15  
## 254 ACH-000255 LMSU  
## 255 ACH-000256 COV318  
## 256 ACH-000257 COR-L279  
## 257 ACH-000258 Du4475  
## 258 ACH-000259 KELLY  
## 259 ACH-000260 SK-N-AS  
## 260 ACH-000261 RERF-LC-AI  
## 261 ACH-000262 UOK101  
## 262 ACH-000263 KASUMI-1  
## 263 ACH-000264 Calu-6  
## 264 ACH-000265 KP-4  
## 265 ACH-000266 SNU-213  
## 266 ACH-000267 HDLM-2  
## 267 ACH-000268 SNU-245  
## 268 ACH-000269 AM-38  
## 269 ACH-000270 HPAC  
## 270 ACH-000271 SU-DHL-10  
## 271 ACH-000272 SLR 24  
## 272 ACH-000273 SF539  
## 273 ACH-000274 Hs 852.T  
## 274 ACH-000275 Hs 834.T  
## 275 ACH-000276 HCC38  
## 276 ACH-000277 HCC1419  
## 277 ACH-000278 COV362  
## 278 ACH-000279 EWS502  
## 279 ACH-000280 SNU-840  
## 280 ACH-000281 KP-2  
## 281 ACH-000282 NCI-H1755  
## 282 ACH-000283 A1207  
## 283 ACH-000284 Hs 840.T  
## 284 ACH-000285 Toledo  
## 285 ACH-000286 SNU-1033  
## 286 ACH-000287 NU-DUL-1  
## 287 ACH-000288 BT-549  
## 288 ACH-000289 SNU-466  
## 289 ACH-000290 NCI-H209  
## 290 ACH-000291 OV-90  
## 291 ACH-000292 NCI-H841  
## 292 ACH-000293 KLE  
## 293 ACH-000294 NB-4  
## 294 ACH-000295 EM-2  
## 295 ACH-000296 OUMS-23  
## 296 ACH-000297 NCI-H889  
## 297 ACH-000298 NCI-H2029  
## 298 ACH-000299 HNT-34  
## 299 ACH-000300 SLR 25  
## 300 ACH-000301 LAMA-84  
## 301 ACH-000302 SNU-1077  
## 302 ACH-000303 SNU-5  
## 303 ACH-000304 WM-115  
## 304 ACH-000305 EC-GI-10  
## 305 ACH-000306 Hs 688(A).T  
## 306 ACH-000307 PK-1  
## 307 ACH-000308 EFO-21  
## 308 ACH-000309 SK-LU-1  
## 309 ACH-000310 IMR-32  
## 310 ACH-000311 NCI-H2122  
## 311 ACH-000312 SK-N-BE(2)  
## 312 ACH-000313 KMRC-3  
## 313 ACH-000314 HCC-2108  
## 314 ACH-000315 KARPAS-422  
## 315 ACH-000316 SNU-886  
## 316 ACH-000317 TUHR14TKB  
## 317 ACH-000318 TE-10  
## 318 ACH-000319 MPP 89  
## 319 ACH-000320 PSN1  
## 320 ACH-000321 MOLM-6  
## 321 ACH-000322 HT-144  
## 322 ACH-000323 42-MG-BA  
## 323 ACH-000324 JHOC-5  
## 324 ACH-000325 SNU-620  
## 325 ACH-000326 JURL-MK1  
## 326 ACH-000327 NCI-H1395  
## 327 ACH-000328 LN-215  
## 328 ACH-000329 CCF-STTG1  
## 329 ACH-000330 EFM-19  
## 330 ACH-000331 IST-MES2  
## 331 ACH-000332 YAPC  
## 332 ACH-000333 JHOM-2B  
## 333 ACH-000334 DB  
## 334 ACH-000335 MSTO-211H  
## 335 ACH-000336 OCI-AML3  
## 336 ACH-000337 NCI-H3122  
## 337 ACH-000338 SR-786  
## 338 ACH-000339 HCC-461  
## 339 ACH-000340 Hs 870.T  
## 340 ACH-000341 SK-N-FI  
## 341 ACH-000342 CL-14  
## 342 ACH-000343 NCI-H522  
## 343 ACH-000344 SNU-668  
## 344 ACH-000345 KP-N-RT-BM-1  
## 345 ACH-000346 JVM-3  
## 346 ACH-000347 QGP-1  
## 347 ACH-000348 RPMI-7951  
## 348 ACH-000349 HCC1500  
## 349 ACH-000350 COLO-678  
## 350 ACH-000351 MKN1  
## 351 ACH-000352 HCC1428  
## 352 ACH-000353 TE-15  
## 353 ACH-000354 Capan-1  
## 354 ACH-000355 NCI-H82  
## 355 ACH-000356 MKN-45  
## 356 ACH-000357 JeKo-1  
## 357 ACH-000358 NCI-H69  
## 358 ACH-000359 MG-63  
## 359 ACH-000360 NCI-H508  
## 360 ACH-000361 SK-HEP-1  
## 361 ACH-000362 MOLM-13  
## 362 ACH-000363 SK-MM-2  
## 363 ACH-000364 U-2 OS  
## 364 ACH-000365 SU-DHL-4  
## 365 ACH-000366 SK-N-DZ  
## 366 ACH-000367 NCI-H226  
## 367 ACH-000368 SNU-1105  
## 368 ACH-000369 MOLM-16  
## 369 ACH-000370 SNU-626  
## 370 ACH-000371 RL  
## 371 ACH-000372 P12-ICHIKAWA  
## 372 ACH-000373 SKM-1  
## 373 ACH-000374 HCC1143  
## 374 ACH-000375 G-402  
## 375 ACH-000376 SF-295  
## 376 ACH-000377 SNU-478  
## 377 ACH-000378 NCI-H647  
## 378 ACH-000379 NCI-H1781  
## 379 ACH-000380 KMS-12-BM  
## 380 ACH-000381 T84  
## 381 ACH-000382 COR-L24  
## 382 ACH-000383 OE33  
## 383 ACH-000384 SW 780  
## 384 ACH-000385 SK-RC-20  
## 385 ACH-000386 KG-1  
## 386 ACH-000387 TF-1  
## 387 ACH-000388 NU-DHL-1  
## 388 ACH-000389 H4  
## 389 ACH-000390 LUDLU-1  
## 390 ACH-000391 MHH-ES-1  
## 391 ACH-000392 Calu-3  
## 392 ACH-000393 HLF  
## 393 ACH-000394 NCI-H2081  
## 394 ACH-000395 NCI-H520  
## 395 ACH-000396 J82  
## 396 ACH-000397 TEN  
## 397 ACH-000398 RI-1  
## 398 ACH-000399 NCI-H2196  
## 399 ACH-000400 SK-CO-1  
## 400 ACH-000401 COLO-800  
## 401 ACH-000402 BL-70  
## 402 ACH-000403 NCI-H747  
## 403 ACH-000404 K029AX  
## 404 ACH-000405 MEC-1  
## 405 ACH-000406 U-937  
## 406 ACH-000407 SNU-685  
## 407 ACH-000408 TE-5  
## 408 ACH-000409 OVSAHO  
## 409 ACH-000410 Saos-2  
## 410 ACH-000411 769-P  
## 411 ACH-000412 SNU-1197  
## 412 ACH-000413 Hs 739.T  
## 413 ACH-000414 NCI-H1944  
## 414 ACH-000415 BICR 6  
## 415 ACH-000416 NCI-H838  
## 416 ACH-000417 Panc 08.13  
## 417 ACH-000418 SW 1353  
## 418 ACH-000419 KMS-28BM  
## 419 ACH-000420 SNU-449  
## 420 ACH-000421 SW837  
## 421 ACH-000422 SNU-475  
## 422 ACH-000423 SK-MEL-3  
## 423 ACH-000424 TC-71  
## 424 ACH-000425 UACC-62  
## 425 ACH-000426 KMS-20  
## 426 ACH-000427 NCI-N87  
## 427 ACH-000428 UO-31  
## 428 ACH-000429 A-704  
## 429 ACH-000430 TYK-nu  
## 430 ACH-000431 NCI-H1694  
## 431 ACH-000432 BV-173  
## 432 ACH-000433 Caki-1  
## 433 ACH-000434 NCI-H1915  
## 434 ACH-000435 EFE-184  
## 435 ACH-000436 OCI-My7  
## 436 ACH-000437 SW 1088  
## 437 ACH-000438 Lu-65  
## 438 ACH-000439 ME-1  
## 439 ACH-000440 CA46  
## 440 ACH-000441 SH-4  
## 441 ACH-000442 RERF-LC-Sq1  
## 442 ACH-000443 OVKATE  
## 443 ACH-000444 LU99  
## 444 ACH-000445 KNS-60  
## 445 ACH-000446 KP-N-SI9s  
## 446 ACH-000447 NCI-H2228  
## 447 ACH-000448 NCI-H1666  
## 448 ACH-000449 MES-SA  
## 449 ACH-000450 MEL-HO  
## 450 ACH-000451 NCI-H2085  
## 451 ACH-000452 TE-8  
## 452 ACH-000453 MOLP-2  
## 453 ACH-000454 HCC-95  
## 454 ACH-000455 LN-428  
## 455 ACH-000456 B-CPAP  
## 456 ACH-000457 CAL-54  
## 457 ACH-000458 CJM  
## 458 ACH-000459 TUHR10TKB  
## 459 ACH-000460 SNU-8  
## 460 ACH-000461 SNU-1196  
## 461 ACH-000462 NALM-1  
## 462 ACH-000463 NCI-H460  
## 463 ACH-000464 CAS-1  
## 464 ACH-000465 SK-MEL-1  
## 465 ACH-000466 SNU-216  
## 466 ACH-000467 HCC-56  
## 467 ACH-000468 PK-45H  
## 468 ACH-000469 YH-13  
## 469 ACH-000470 SW1463  
## 470 ACH-000471 Li-7  
## 471 ACH-000472 HSC-2  
## 472 ACH-000473 RT-112  
## 473 ACH-000475 huH-1  
## 474 ACH-000476 JHH-4  
## 475 ACH-000477 Malme-3M  
## 476 ACH-000478 SNU-387  
## 477 ACH-000479 KNS-81  
## 478 ACH-000480 HuH-7  
## 479 ACH-000481 NCI-H2170  
## 480 ACH-000482 RERF-LC-KJ  
## 481 ACH-000483 SNU-182  
## 482 ACH-000484 VMRC-RCW  
## 483 ACH-000485 GSU  
## 484 ACH-000486 KU-19-19  
## 485 ACH-000487 F-36P  
## 486 ACH-000488 TE-11  
## 487 ACH-000489 SW1116  
## 488 ACH-000490 SF767  
## 489 ACH-000491 NCI-H716  
## 490 ACH-000492 MUTZ-5  
## 491 ACH-000493 SNU-423  
## 492 ACH-000494 OELE  
## 493 ACH-000495 TUHR4TKB  
## 494 ACH-000496 NCI-H1792  
## 495 ACH-000498 KO52  
## 496 ACH-000499 EW8  
## 497 ACH-000500 SNU-46  
## 498 ACH-000501 LS123  
## 499 ACH-000502 TCC-PAN2  
## 500 ACH-000503 BICR 16  
## 501 ACH-000504 SNB75  
## 502 ACH-000505 RKN  
## 503 ACH-000506 NCI-H146  
## 504 ACH-000507 KE-39  
## 505 ACH-000508 COR-L88  
## 506 ACH-000509 HuT 78  
## 507 ACH-000510 NCI-H1299  
## 508 ACH-000511 Calu-1  
## 509 ACH-000512 INA6  
## 510 ACH-000513 SNU-1272  
## 511 ACH-000514 NCI-H1092  
## 512 ACH-000515 HCC-33  
## 513 ACH-000516 CAL-78  
## 514 ACH-000517 SNU-410  
## 515 ACH-000518 CAL-33  
## 516 ACH-000519 PEER  
## 517 ACH-000520 59M  
## 518 ACH-000521 NCI-H2030  
## 519 ACH-000522 UM-UC-3  
## 520 ACH-000523 NCI-H1184  
## 521 ACH-000524 KURAMOCHI  
## 522 ACH-000525 NCI-H2171  
## 523 ACH-000526 Hs 821.T  
## 524 ACH-000527 OVISE  
## 525 ACH-000528 ABC-1  
## 526 ACH-000529 T1-73  
## 527 ACH-000530 DMS 114  
## 528 ACH-000531 RS-5  
## 529 ACH-000532 SNU-61  
## 530 ACH-000533 NCI-H2004 RT  
## 531 ACH-000534 WSU-DLCL2  
## 532 ACH-000535 BxPC-3  
## 533 ACH-000536 BT-20  
## 534 ACH-000537 SNU-761  
## 535 ACH-000538 HUTU80  
## 536 ACH-000539 Hs 618.T  
## 537 ACH-000540 Hs 606.T  
## 538 ACH-000541 KMS-34  
## 539 ACH-000542 Hey-A8  
## 540 ACH-000543 SNU-489  
## 541 ACH-000544 OE21  
## 542 ACH-000545 VM-CUB1  
## 543 ACH-000546 HSC-4  
## 544 ACH-000547 HT-1197  
## 545 ACH-000548 BHY  
## 546 ACH-000549 SNU-1076  
## 547 ACH-000550 IGR-39  
## 548 ACH-000551 K-562  
## 549 ACH-000552 HT-29  
## 550 ACH-000553 Sq-1  
## 551 ACH-000554 UACC-893  
## 552 ACH-000555 A-498  
## 553 ACH-000556 SIHA  
## 554 ACH-000557 AML-193  
## 555 ACH-000558 A-172  
## 556 ACH-000559 NCI-H1836  
## 557 ACH-000560 ECC10  
## 558 ACH-000561 T.T  
## 559 ACH-000562 HCC-78  
## 560 ACH-000563 EBC-1  
## 561 ACH-000564 KHM-1B  
## 562 ACH-000565 RCM-1  
## 563 ACH-000566 SW-1710  
## 564 ACH-000567 ST486  
## 565 ACH-000568 UACC-812  
## 566 ACH-000569 IST-MES1  
## 567 ACH-000570 YKG1  
## 568 ACH-000571 T98G  
## 569 ACH-000572 G-361  
## 570 ACH-000573 MDA-MB-436  
## 571 ACH-000574 FU-OV-1  
## 572 ACH-000575 HCC-364  
## 573 ACH-000576 KMS-27  
## 574 ACH-000577 JHH-2  
## 575 ACH-000578 HCC-1171  
## 576 ACH-000579 UACC-257  
## 577 ACH-000580 C32  
## 578 ACH-000581 SNU-16  
## 579 ACH-000582 COLO 741  
## 580 ACH-000583 MC116  
## 581 ACH-000584 JHOS-4  
## 582 ACH-000585 EPLC-272H  
## 583 ACH-000586 NCI-H1876  
## 584 ACH-000587 NCI-H1975  
## 585 ACH-000588 KMS-26  
## 586 ACH-000589 NCI-H1437  
## 587 ACH-000590 NCI-H2073  
## 588 ACH-000591 LN-235  
## 589 ACH-000592 TM-31  
## 590 ACH-000593 BC-3C  
## 591 ACH-000594 DMS 153  
## 592 ACH-000595 LN-229  
## 593 ACH-000596 LCLC-97TM1  
## 594 ACH-000597 TTC-709  
## 595 ACH-000598 KMS-21BM  
## 596 ACH-000599 PA-TU-8902  
## 597 ACH-000600 SLR 26  
## 598 ACH-000601 MIA PaCa-2  
## 599 ACH-000602 M-07e  
## 600 ACH-000603 BEN  
## 601 ACH-000604 KYO-1  
## 602 ACH-000605 TE-6  
## 603 ACH-000606 PE/CA-PJ34 (clone C12)  
## 604 ACH-000607 KYM-1  
## 605 ACH-000608 COV644  
## 606 ACH-000609 SF126  
## 607 ACH-000610 NCI-H2227  
## 608 ACH-000611 SU-DHL-6  
## 609 ACH-000612 HuT 102  
## 610 ACH-000613 HOS  
## 611 ACH-000614 RVH-421  
## 612 ACH-000615 SK-MEL-28  
## 613 ACH-000616 Hs 746T  
## 614 ACH-000617 OVCAR-4  
## 615 ACH-000618 SNU-1041  
## 616 ACH-000619 PE/CA-PJ15  
## 617 ACH-000620 JHH-1  
## 618 ACH-000621 MDA-MB-157  
## 619 ACH-000622 KNS-42  
## 620 ACH-000623 SNU-201  
## 621 ACH-000624 HCC1806  
## 622 ACH-000625 Hep 3B2.1-7  
## 623 ACH-000626 U266B1  
## 624 ACH-000627 LCLC-103H  
## 625 ACH-000628 NCI-H596  
## 626 ACH-000629 IOMM-Lee  
## 627 ACH-000630 YD-8  
## 628 ACH-000631 KS-1  
## 629 ACH-000632 Hs 944.T  
## 630 ACH-000633 FU97  
## 631 ACH-000634 LN-340  
## 632 ACH-000635 SNU-119  
## 633 ACH-000636 RPMI-8402  
## 634 ACH-000637 KYSE-520  
## 635 ACH-000638 NCI-H441  
## 636 ACH-000639 NCI-H211  
## 637 ACH-000640 SK-MEL-31  
## 638 ACH-000641 CMK  
## 639 ACH-000642 HMEL  
## 640 ACH-000643 HDQ-P1  
## 641 ACH-000644 COLO 829  
## 642 ACH-000645 JL-1  
## 643 ACH-000646 OVMANA  
## 644 ACH-000647 TE-1  
## 645 ACH-000648 NCI-H28  
## 646 ACH-000649 786-O  
## 647 ACH-000650 IGR-37  
## 648 ACH-000651 SW 620  
## 649 ACH-000652 SUIT-2  
## 650 ACH-000653 JJN-3  
## 651 ACH-000654 Raji  
## 652 ACH-000655 SF268  
## 653 ACH-000656 SU-DHL-8  
## 654 ACH-000657 A2780  
## 655 ACH-000658 KMS-18  
## 656 ACH-000659 SCLC-21H  
## 657 ACH-000660 SU-DHL-5  
## 658 ACH-000661 WM1799  
## 659 ACH-000662 COR-L23  
## 660 ACH-000663 OVTOKO  
## 661 ACH-000664 SU-DHL-1  
## 662 ACH-000665 SK-MES-1  
## 663 ACH-000666 NCI-H1355  
## 664 ACH-000667 HCC-44  
## 665 ACH-000668 HCC70  
## 666 ACH-000669 SW 900  
## 667 ACH-000670 SBC-5  
## 668 ACH-000671 HuH-6  
## 669 ACH-000672 IA-LM  
## 670 ACH-000673 LN-443  
## 671 ACH-000674 NUGC-4  
## 672 ACH-000675 NCI-H1734  
## 673 ACH-000676 LN-464  
## 674 ACH-000677 SW 1573  
## 675 ACH-000678 MKN7  
## 676 ACH-000679 OE-19  
## 677 ACH-000680 SW948  
## 678 ACH-000681 A549  
## 679 ACH-000682 SNU-1066  
## 680 ACH-000683 SNU-503  
## 681 ACH-000684 KMRC-1  
## 682 ACH-000685 L3.3  
## 683 ACH-000686 SNU-878  
## 684 ACH-000687 CI-1  
## 685 ACH-000688 OV7  
## 686 ACH-000689 RH-18  
## 687 ACH-000690 HCC-2814  
## 688 ACH-000691 HCC2157  
## 689 ACH-000692 SNU-899  
## 690 ACH-000693 KYSE-180  
## 691 ACH-000694 TE-9  
## 692 ACH-000695 COR-L47  
## 693 ACH-000696 OVCAR-8  
## 694 ACH-000697 A3/KAW  
## 695 ACH-000698 DMS 53  
## 696 ACH-000699 HCC1395  
## 697 ACH-000700 NCI-H2882  
## 698 ACH-000701 RMUG-S  
## 699 ACH-000702 L-1236  
## 700 ACH-000703 DMS 79  
## 701 ACH-000704 OAW42  
## 702 ACH-000705 LC-1F  
## 703 ACH-000706 EKVX  
## 704 ACH-000707 P3HR-1  
## 705 ACH-000708 SNU-283  
## 706 ACH-000709 KMRC-2  
## 707 ACH-000710 NCI-H854  
## 708 ACH-000711 JIMT-1  
## 709 ACH-000712 HCC-1833  
## 710 ACH-000713 Caov-3  
## 711 ACH-000714 KMS-11  
## 712 ACH-000715 SNU-1214  
## 713 ACH-000716 TT2609-C02  
## 714 ACH-000717 COLO-680N  
## 715 ACH-000718 NCI-H2291  
## 716 ACH-000719 RMG-I  
## 717 ACH-000720 TCCSUP  
## 718 ACH-000721 HMC-1-8  
## 719 ACH-000722 SNU-C1  
## 720 ACH-000723 YD-10B  
## 721 ACH-000724 HT-1376  
## 722 ACH-000725 HCC202  
## 723 ACH-000726 TE-14  
## 724 ACH-000727 NCI-H2066  
## 725 ACH-000728 KASUMI-2  
## 726 ACH-000729 NCI-H1963  
## 727 ACH-000730 SK-MEL-5  
## 728 ACH-000731 HCC-2279  
## 729 ACH-000732 PE/CA-PJ41 (clone D2)  
## 730 ACH-000733 NCI-H1838  
## 731 ACH-000734 JHH-5  
## 732 ACH-000735 PE/CA-PJ49  
## 733 ACH-000736 SNU-601  
## 734 ACH-000737 NCI-H1385  
## 735 ACH-000738 GB-1  
## 736 ACH-000739 Hep G2  
## 737 ACH-000740 A-253  
## 738 ACH-000741 U-BLC1  
## 739 ACH-000742 DM-3  
## 740 ACH-000743 COR-L95  
## 741 ACH-000744 NCI-H1623  
## 742 ACH-000745 MOLP-8  
## 743 ACH-000746 GSS  
## 744 ACH-000747 NCI-H1703  
## 745 ACH-000748 SJSA-1  
## 746 ACH-000749 DMS 273  
## 747 ACH-000750 LOX IMVI  
## 748 ACH-000751 OCI-M1  
## 749 ACH-000752 NCI-H196  
## 750 ACH-000753 JMSU-1  
## 751 ACH-000754 L-428  
## 752 ACH-000755 HCC2218  
## 753 ACH-000756 GI-1  
## 754 ACH-000757 A427  
## 755 ACH-000758 MKN74  
## 756 ACH-000759 MDA-MB-175-VII  
## 757 ACH-000760 LNZ308  
## 758 ACH-000761 NUGC-2  
## 759 ACH-000762 YD-38  
## 760 ACH-000763 MM1-S  
## 761 ACH-000764 SH-10-TC  
## 762 ACH-000765 WM-983B  
## 763 ACH-000766 NCI-H1648  
## 764 ACH-000767 NCI-H526  
## 765 ACH-000768 MDA-MB-231  
## 766 ACH-000769 LK-2  
## 767 ACH-000770 P31/FUJ  
## 768 ACH-000771 BICR 56  
## 769 ACH-000772 TE 441.T  
## 770 ACH-000773 Ki-JK  
## 771 ACH-000774 RERF-LC-Ad2  
## 772 ACH-000775 NCI-H727  
## 773 ACH-000776 ONS-76  
## 774 ACH-000777 KYSE-30  
## 775 ACH-000778 HSC-3  
## 776 ACH-000779   
## 777 ACH-000780 NCI-H1105  
## 778 ACH-000781 NCI-H2023  
## 779 ACH-000782 SEM  
## 780 ACH-000783 CAMA-1  
## 781 ACH-000784 KYSE-70  
## 782 ACH-000785 NCI-H2126  
## 783 ACH-000786 Daudi  
## 784 ACH-000787 LXF-289  
## 785 ACH-000788 A2058  
## 786 ACH-000789 NCI-H810  
## 787 ACH-000790 SHP-77  
## 788 ACH-000791 RERF-LC-Ad1  
## 789 ACH-000792 BFTC-909  
## 790 ACH-000793 KATO III  
## 791 ACH-000794 BICR 22  
## 792 ACH-000795 MOLT-13  
## 793 ACH-000796 MCAS  
## 794 ACH-000797 HLF-a  
## 795 ACH-000798 CL-40  
## 796 ACH-000799 Hs 695T  
## 797 ACH-000800 NCI-H446  
## 798 ACH-000801 Hs 936.T  
## 799 ACH-000802 BFTC-905  
## 800 ACH-000803 COLO 668  
## 801 ACH-000804 NB-1  
## 802 ACH-000805 COLO-679  
## 803 ACH-000806 L-540  
## 804 ACH-000807 SNU-738  
## 805 ACH-000808 HuH28  
## 806 ACH-000809 KYSE-410  
## 807 ACH-000810 SK-MEL-30  
## 808 ACH-000811 SK-OV-3  
## 809 ACH-000812 COLO-783  
## 810 ACH-000813 T3M-10  
## 811 ACH-000814 Hs 939.T  
## 812 ACH-000815 KM-H2  
## 813 ACH-000816 NCI-H524  
## 814 ACH-000817 RPMI 8226  
## 815 ACH-000818 BT-483  
## 816 ACH-000819 LN-18  
## 817 ACH-000820 SW403  
## 818 ACH-000821 EJM  
## 819 ACH-000822 SK-MEL-24  
## 820 ACH-000823 KYSE-140  
## 821 ACH-000824 KYSE-510  
## 822 ACH-000825 HOP-92  
## 823 ACH-000826 CAL-12T  
## 824 ACH-000827 WM-793  
## 825 ACH-000828 ZR-75-30  
## 826 ACH-000829 HuNS1  
## 827 ACH-000830 NCI-H1436  
## 828 ACH-000831 HEC-50B  
## 829 ACH-000832 CAL 27  
## 830 ACH-000833 RH-30  
## 831 ACH-000834 UM-UC-1  
## 832 ACH-000835 GCT  
## 833 ACH-000836 YD-15  
## 834 ACH-000837 NCI-H322  
## 835 ACH-000838 AMO-1  
## 836 ACH-000839 SCaBER  
## 837 ACH-000840 HCC-366  
## 838 ACH-000841 NCI-H2087  
## 839 ACH-000842 SW 480  
## 840 ACH-000843 HARA  
## 841 ACH-000844 DMS 454  
## 842 ACH-000845 NCI-H1373  
## 843 ACH-000846 FaDu  
## 844 ACH-000847 HGC-27  
## 845 ACH-000848 JHH-7  
## 846 ACH-000849 MDA-MB-468  
## 847 ACH-000850 Hs 698.T  
## 848 ACH-000851 MOR/CPR  
## 849 ACH-000852 NCI-H1435  
## 850 ACH-000853 NCI-H661  
## 851 ACH-000854 OCI-MY5  
## 852 ACH-000855 KYSE-150  
## 853 ACH-000856 CAL-51  
## 854 ACH-000857 CAL-85-1  
## 855 ACH-000858 KNS-62  
## 856 ACH-000859 HCC1954  
## 857 ACH-000860 NCI-H358  
## 858 ACH-000861 HOP-62  
## 859 ACH-000862 KMBC-2  
## 860 ACH-000863 DBTRG-05MG  
## 861 ACH-000864 COLO 684  
## 862 ACH-000865 KYSE-450  
## 863 ACH-000866 NCI-H1048  
## 864 ACH-000867 ChaGo-K-1  
## 865 ACH-000868 HCC-1195  
## 866 ACH-000869 NCI-H1568  
## 867 ACH-000870 NCI-H1930  
## 868 ACH-000871 NCI-H510  
## 869 ACH-000872 HCC515  
## 870 ACH-000873 KYSE-270  
## 871 ACH-000874 RS4;11  
## 872 ACH-000875 NCI-H2347  
## 873 ACH-000876 MDA-MB-415  
## 874 ACH-000877 EB1  
## 875 ACH-000878 HCC-15  
## 876 ACH-000879 MFE-296  
## 877 ACH-000880 AGS  
## 878 ACH-000881 MEL-JUSO  
## 879 ACH-000882 IGR-1  
## 880 ACH-000883 SW 1783  
## 881 ACH-000884 MDA-MB-435S  
## 882 ACH-000885 TOV-21G  
## 883 ACH-000886 NCI-H2009  
## 884 ACH-000887 SF-172  
## 885 ACH-000888 NCI-H1793  
## 886 ACH-000889 KMM-1  
## 887 ACH-000890 SW 1271  
## 888 ACH-000891 HCC-1438  
## 889 ACH-000892 NCI-H1563  
## 890 ACH-000893 NCI-H1651  
## 891 ACH-000894 NCI-H1869  
## 892 ACH-000895 CL-34  
## 893 ACH-000896 647-V  
## 894 ACH-000897 FTC-238  
## 895 ACH-000898 SNU-719  
## 896 ACH-000899 WM-88  
## 897 ACH-000900 NCI-H23  
## 898 ACH-000901 HCC-1359  
## 899 ACH-000902 CAL-148  
## 900 ACH-000903 FTC-133  
## 901 ACH-000904 NCI-H2106  
## 902 ACH-000905 5637  
## 903 ACH-000906 ES-2  
## 904 ACH-000907 SNU-349  
## 905 ACH-000908 SNU-520  
## 906 ACH-000909 JHUEM-2  
## 907 ACH-000910 MDA-MB-453  
## 908 ACH-000911 NUGC-3  
## 909 ACH-000912 NCI-H2286  
## 910 ACH-000913 ESS-1  
## 911 ACH-000914 HT  
## 912 ACH-000915 IPC-298  
## 913 ACH-000916 NCI-H1573  
## 914 ACH-000917 TE-4  
## 915 ACH-000918 MOLT-16  
## 916 ACH-000919 IM95  
## 917 ACH-000920 CML-T1  
## 918 ACH-000921 NCI-H157-DM  
## 919 ACH-000922 RCH-ACV  
## 920 ACH-000923 BCP-1  
## 921 ACH-000924 NCI-H2172  
## 922 ACH-000925 DV-90  
## 923 ACH-000926 HT-55  
## 924 ACH-000927 BT-474  
## 925 ACH-000928 JHUEM-1  
## 926 ACH-000929 NCI-H2110  
## 927 ACH-000930 HCC1569  
## 928 ACH-000931 HMCB  
## 929 ACH-000932 SNU-1  
## 930 ACH-000933 SNU-324  
## 931 ACH-000934 MDA-MB-361  
## 932 ACH-000935 MDST8  
## 933 ACH-000936 EFO-27  
## 934 ACH-000937 PF-382  
## 935 ACH-000938 NALM-6  
## 936 ACH-000939 SK-UT-1  
## 937 ACH-000940 AN3 CA  
## 938 ACH-000941 HEC-1-B  
## 939 ACH-000942 HPB-ALL  
## 940 ACH-000943 RKO  
## 941 ACH-000944 NAMALWA  
## 942 ACH-000945 NCI-H650  
## 943 ACH-000946 HEC-265  
## 944 ACH-000947 OVK18  
## 945 ACH-000948 23132/87  
## 946 ACH-000949 TGBC11TKB  
## 947 ACH-000950 LoVo  
## 948 ACH-000951 NCI-H2342  
## 949 ACH-000952 MDA PCa 2b  
## 950 ACH-000953 SUP-T1  
## 951 ACH-000954 HEC-1-A  
## 952 ACH-000955 SNU-407  
## 953 ACH-000956 22Rv1  
## 954 ACH-000957 LS 180  
## 955 ACH-000958 SW48  
## 956 ACH-000959 SNU-C4  
## 957 ACH-000960 Reh  
## 958 ACH-000961 Ishikawa (Heraklio) 02 ER-  
## 959 ACH-000962 OC 314  
## 960 ACH-000963 CCK-81  
## 961 ACH-000964 MOLT-3  
## 962 ACH-000965 RL95-2  
## 963 ACH-000966 IGROV1  
## 964 ACH-000967 SNU-C2A  
## 965 ACH-000968 COLO 792  
## 966 ACH-000969 KM12  
## 967 ACH-000970 SNU-C5  
## 968 ACH-000971 HCT 116  
## 969 ACH-000972 HEC-151  
## 970 ACH-000973 639-V  
## 971 ACH-000974 SNG-M  
## 972 ACH-000975 HCC2450  
## 973 ACH-000976 HuCCT1  
## 974 ACH-000977 LNCaP clone FGC  
## 975 ACH-000978 EN  
## 976 ACH-000979 DU 145  
## 977 ACH-000980 NCI-H1155  
## 978 ACH-000981 DND-41  
## 979 ACH-000982 GP2d  
## 980 ACH-000983 KCL-22  
## 981 ACH-000984 HEC-6  
## 982 ACH-000985 LS411N  
## 983 ACH-000986 HT115  
## 984 ACH-000987 MeWo  
## 985 ACH-000988 MFE-319  
## 986 ACH-000989 SNU-175  
## 987 ACH-000990 HEC-108  
## 988 ACH-000991 SNU-81  
## 989 ACH-000992 BICR 18  
## 990 ACH-000993 JHUEM-7  
## 991 ACH-000994 HEC-59  
## 992 ACH-000995 JURKAT  
## 993 ACH-000996 HEC-251  
## 994 ACH-000997 HCT-15  
## 995 ACH-000998 CW-2  
## 996 ACH-000999 SNU-1040  
## 997 ACH-001000 1321N1  
## 998 ACH-001001 143B  
## 999 ACH-001002 451Lu  
## 1000 ACH-001007 A673STAG2KO16  
## 1001 ACH-001008 A673STAG2KO45  
## 1002 ACH-001009 A673STAG2NT14  
## 1003 ACH-001010 A673STAG2NT23  
## 1004 ACH-001015 AZ521  
## 1005 ACH-001016 Becker  
## 1006 ACH-001017 BGC823  
## 1007 ACH-001018 BJ hTERT  
## 1008 ACH-001020 BT-16  
## 1009 ACH-001021 C3A  
## 1010 ACH-001022 CBAGPN  
## 1011 ACH-001023 CGTH-W-1  
## 1012 ACH-001024 CHL-1  
## 1013 ACH-001028 CHLA-06  
## 1014 ACH-001029 CHLA-10  
## 1015 ACH-001030 CHLA-218  
## 1016 ACH-001031 CHLA-266  
## 1017 ACH-001032 CHLA-32  
## 1018 ACH-001033 CHLA-57  
## 1019 ACH-001034 CHLA-9  
## 1020 ACH-001035 CHLA-99  
## 1021 ACH-001036 CMK-11-5  
## 1022 ACH-001037 CMK-86  
## 1023 ACH-001038 COG-E-352  
## 1024 ACH-001039 COLO 205  
## 1025 ACH-001041 CHL-1-DM  
## 1026 ACH-001042 COLO-704  
## 1027 ACH-001043 COLO 775  
## 1028 ACH-001044 COLO 818  
## 1029 ACH-001045 COLO 849  
## 1030 ACH-001047 COR-L51  
## 1031 ACH-001048 COV504  
## 1032 ACH-001049 CPC-N  
## 1033 ACH-001050 CW9019  
## 1034 ACH-001052 D384  
## 1035 ACH-001053 D425  
## 1036 ACH-001054 D458  
## 1037 ACH-001055 D556  
## 1038 ACH-001057 DERL-2  
## 1039 ACH-001059 DL  
## 1040 ACH-001060 DL-40  
## 1041 ACH-001061 DLD-1  
## 1042 ACH-001063 DOV13  
## 1043 ACH-001064 EB-2  
## 1044 ACH-001065 Evsa-T  
## 1045 ACH-001066 EWS-834  
## 1046 ACH-001067 F5  
## 1047 ACH-001068 FE-PD  
## 1048 ACH-001071 GLC-82  
## 1049 ACH-001072 GR-M  
## 1050 ACH-001075 NCI-H292  
## 1051 ACH-001079 HCC1897  
## 1052 ACH-001081 HCC2998  
## 1053 ACH-001084 HCT-8  
## 1054 ACH-001086 HeLa  
## 1055 ACH-001087 HK-2  
## 1056 ACH-001088 HLC-1  
## 1057 ACH-001089 HLE  
## 1058 ACH-001090 HN  
## 1059 ACH-001091 HRT18  
## 1060 ACH-001093 Hs 604.T  
## 1061 ACH-001094 HTK-  
## 1062 ACH-001096 JR  
## 1063 ACH-001097 KARPAS384  
## 1064 ACH-001098 KCI-MOH1  
## 1065 ACH-001099 KD  
## 1066 ACH-001100 KHYG-1  
## 1067 ACH-001101 KLM-1  
## 1068 ACH-001106 KOPN-8  
## 1069 ACH-001107 KP-1N  
## 1070 ACH-001109 KP-MRT-RY  
## 1071 ACH-001111 L82  
## 1072 ACH-001113 LC-1/sq-SF  
## 1073 ACH-001118 M059J  
## 1074 ACH-001119 Mac-2A  
## 1075 ACH-001121 MEC2  
## 1076 ACH-001122 MKL-1  
## 1077 ACH-001123 MKL-2  
## 1078 ACH-001125 MOG-G-CCM  
## 1079 ACH-001126 MOG-G-UVW  
## 1080 ACH-001127 MOLT-4  
## 1081 ACH-001128 MON  
## 1082 ACH-001129 MONO-MAC-1  
## 1083 ACH-001130 MOTN-1  
## 1084 ACH-001131 MS-1  
## 1085 ACH-001132 MTA  
## 1086 ACH-001134 MYLA  
## 1087 ACH-001136 NCI-H187  
## 1088 ACH-001137 NCI-H1993  
## 1089 ACH-001138 NCI-H2141  
## 1090 ACH-001142 NHAHTDD  
## 1091 ACH-001143 NKL  
## 1092 ACH-001144 OC315  
## 1093 ACH-001145 OC 316  
## 1094 ACH-001146 OCI-Ly10  
## 1095 ACH-001147 OCI-Ly12  
## 1096 ACH-001148 OCILY-13  
## 1097 ACH-001150 OUMS-27  
## 1098 ACH-001151 OVCAR-5  
## 1099 ACH-001162 PCM6  
## 1100 ACH-001163 CCLF\_PEDS\_0001\_T  
## 1101 ACH-001164 CCLF\_PEDS\_0003\_T  
## 1102 ACH-001170 PeTa  
## 1103 ACH-001171 PL45  
## 1104 ACH-001172 U-251 MG  
## 1105 ACH-001175 RCC-4  
## 1106 ACH-001182 RPMI 6666  
## 1107 ACH-001183 RT112/84  
## 1108 ACH-001184 SCMC-RM2  
## 1109 ACH-001188 SH-SY5Y  
## 1110 ACH-001189 SJRH-30  
## 1111 ACH-001190 SK-MEL-2  
## 1112 ACH-001192 SK-NEP-1  
## 1113 ACH-001193 SK-PN-DW  
## 1114 ACH-001194 SK-RC-31  
## 1115 ACH-001196 SMS-CTR  
## 1116 ACH-001197 SMZ-1  
## 1117 ACH-001198 SNB19  
## 1118 ACH-001199 SNUC2B  
## 1119 ACH-001200 STM91-01  
## 1120 ACH-001201 SU-MB-002  
## 1121 ACH-001203 SUP-HD1  
## 1122 ACH-001205 TC-32  
## 1123 ACH-001206 TTC-466  
## 1124 ACH-001207 TIG-3 TD  
## 1125 ACH-001208 TK-10  
## 1126 ACH-001210 TTC-1240  
## 1127 ACH-001211 TTC-549  
## 1128 ACH-001212 TTC-642  
## 1129 ACH-001214 U-138 MG  
## 1130 ACH-001224 UM-RC-2  
## 1131 ACH-001225   
## 1132 ACH-001227 UPCI-SCC-090  
## 1133 ACH-001228 UPCI-SCC-152  
## 1134 ACH-001229 UPCI-SCC-154  
## 1135 ACH-001230 UT-7  
## 1136 ACH-001232 UW228  
## 1137 ACH-001233 VMRC-LCD  
## 1138 ACH-001234 VMRC-LCP  
## 1139 ACH-001239 WM-266-4  
## 1140 ACH-001249 YMB-1  
## 1141 ACH-001270 1273/99  
## 1142 ACH-001272 Fuji  
## 1143 ACH-001274 SW982  
## 1144 ACH-001275 SYO1  
## 1145 ACH-001277 Yamato  
## 1146 ACH-001278 BIN-67  
## 1147 ACH-001279 SCCOHT-1  
## 1148 ACH-001280 SCS-214  
## 1149 ACH-001282 CHLA258  
## 1150 ACH-001283 TC-106  
## 1151 ACH-001289 COG-AR-359  
## 1152 ACH-001295 Y-79  
## 1153 ACH-001300 CHLA15  
## 1154 ACH-001301 COGN278  
## 1155 ACH-001302 COG-N-305  
## 1156 ACH-001303 NB-1643  
## 1157 ACH-001306 8305C  
## 1158 ACH-001307 8505C  
## 1159 ACH-001310 HA1E  
## 1160 ACH-001318 PLC/PRF/5  
## 1161 ACH-001321 TT  
## 1162 ACH-001322 CME-1  
## 1163 ACH-001328 A-431  
## 1164 ACH-001329 ANGM-CSS  
## 1165 ACH-001331 BICR 10  
## 1166 ACH-001332 BICR 78  
## 1167 ACH-001333 C-33 A  
## 1168 ACH-001334 C-4 I  
## 1169 ACH-001335 C-4 II  
## 1170 ACH-001336 Ca Ski  
## 1171 ACH-001338 CHP-134  
## 1172 ACH-001339 Colo 794  
## 1173 ACH-001340 COV413A  
## 1174 ACH-001341 DoTc2 4510  
## 1175 ACH-001344 GI-ME-N  
## 1176 ACH-001345 GP5d  
## 1177 ACH-001346 H103  
## 1178 ACH-001347 H157  
## 1179 ACH-001350 HTC-C3  
## 1180 ACH-001353 JOPACA-1  
## 1181 ACH-001354 LAN-2  
## 1182 ACH-001355 LAN-6  
## 1183 ACH-001356 MB-1  
## 1184 ACH-001357 MCF 10A  
## 1185 ACH-001358 MDA-MB-330  
## 1186 ACH-001359 ME-180  
## 1187 ACH-001360 MS751  
## 1188 ACH-001362 NCI-H1770  
## 1189 ACH-001363 NCI-H2135  
## 1190 ACH-001364 NCI-H345  
## 1191 ACH-001365 NCI-H847  
## 1192 ACH-001366 NGP  
## 1193 ACH-001367 NMB  
## 1194 ACH-001368 OAC-M5.1  
## 1195 ACH-001369 OCI-C5x  
## 1196 ACH-001370 OCI-P5x  
## 1197 ACH-001373 OV17R  
## 1198 ACH-001374 PA-1 [PA1]  
## 1199 ACH-001375 PACADD-119  
## 1200 ACH-001376 PACADD-135  
## 1201 ACH-001377 PACADD-137  
## 1202 ACH-001378 PACADD-159  
## 1203 ACH-001379 PACADD-161  
## 1204 ACH-001380 PACADD-165  
## 1205 ACH-001382 PACADD-188  
## 1206 ACH-001383 PWR-1E  
## 1207 ACH-001384 RO82-W-1  
## 1208 ACH-001385 RPMI 2650  
## 1209 ACH-001386 SCLC-22H  
## 1210 ACH-001388 SUM-102PT  
## 1211 ACH-001389 SUM-1315MO2  
## 1212 ACH-001390 SUM-149PT  
## 1213 ACH-001391 SUM-159PT  
## 1214 ACH-001392 SUM-185PE  
## 1215 ACH-001393 SUM-190PT  
## 1216 ACH-001394 SUM-229PE  
## 1217 ACH-001395 SUM-44PE  
## 1218 ACH-001396 SUM-52PE  
## 1219 ACH-001397 SUM225CWN  
## 1220 ACH-001398 SW 156  
## 1221 ACH-001399 SW 626  
## 1222 ACH-001400 SW 954  
## 1223 ACH-001401 SW-13  
## 1224 ACH-001402 SW756  
## 1225 ACH-001403 TO14  
## 1226 ACH-001407 UM-UC-13  
## 1227 ACH-001408 UM-UC-14  
## 1228 ACH-001409 UM-UC-16  
## 1229 ACH-001410 UM-UC-4  
## 1230 ACH-001411 UM-UC-5  
## 1231 ACH-001412 UM-UC-10  
## 1232 ACH-001413 UM-UC-11  
## 1233 ACH-001414 UM-UC-6  
## 1234 ACH-001415 UM-UC7  
## 1235 ACH-001416 UM-UC9  
## 1236 ACH-001417 UMC-11  
## 1237 ACH-001418 UWB1.289  
## 1238 ACH-001419 VP229  
## 1239 ACH-001421 WERI-Rb-1  
## 1240 ACH-001422 WPE1-NA22  
## 1241 ACH-001430 TC138  
## 1242 ACH-001431 TC205  
## 1243 ACH-001433 CCLF\_PEDS\_0008\_T  
## 1244 ACH-001441 92-1  
## 1245 ACH-001442 A388  
## 1246 ACH-001443 ASH-3  
## 1247 ACH-001450 BLUE-1  
## 1248 ACH-001451 BOKU  
## 1249 ACH-001452 BONNA-12  
## 1250 ACH-001453 BPH-1  
## 1251 ACH-001454 C10  
## 1252 ACH-001456 C125PM  
## 1253 ACH-001458 C75  
## 1254 ACH-001459 C80  
## 1255 ACH-001460 C84  
## 1256 ACH-001461 C99  
## 1257 ACH-001481 CHLA-90  
## 1258 ACH-001484 CI  
## 1259 ACH-001485 CII  
## 1260 ACH-001489 COR-L32  
## 1261 ACH-001490 COR-L321  
## 1262 ACH-001494 EGI-1  
## 1263 ACH-001495 EMTOKA  
## 1264 ACH-001496 ESO26  
## 1265 ACH-001497 ESO51  
## 1266 ACH-001498 Farage  
## 1267 ACH-001500 FLO-1  
## 1268 ACH-001509 H357  
## 1269 ACH-001510 H376  
## 1270 ACH-001511 H413  
## 1271 ACH-001513 HCA-1  
## 1272 ACH-001514 HCC1008  
## 1273 ACH-001515 HCS-2  
## 1274 ACH-001516 HCSC-1  
## 1275 ACH-001517 HEC-1  
## 1276 ACH-001518 HEC-116  
## 1277 ACH-001519 H-EMC-SS  
## 1278 ACH-001520 HG-3  
## 1279 ACH-001521 HKA-1  
## 1280 ACH-001522 HMY-1  
## 1281 ACH-001523 HSC-1  
## 1282 ACH-001524 HSC-5  
## 1283 ACH-001525 HT-3  
## 1284 ACH-001526 HuO9  
## 1285 ACH-001528 IHH-4  
## 1286 ACH-001529 JAR  
## 1287 ACH-001530 JEG-3  
## 1288 ACH-001532 JMU-RTK-2  
## 1289 ACH-001533 KARPAS 1718  
## 1290 ACH-001536 KKU-100  
## 1291 ACH-001538 KKU-213  
## 1292 ACH-001539 KML-1  
## 1293 ACH-001540 KMLS-1  
## 1294 ACH-001541 KMS-28PE  
## 1295 ACH-001542 KON  
## 1296 ACH-001543 KOSC-2  
## 1297 ACH-001544 KYAE-1  
## 1298 ACH-001547 LO68  
## 1299 ACH-001548 LS  
## 1300 ACH-001549 Lu-135  
## 1301 ACH-001550 MCC13  
## 1302 ACH-001551 MCC14/2  
## 1303 ACH-001552 MCC26  
## 1304 ACH-001554 mel-202  
## 1305 ACH-001555 Mero-14  
## 1306 ACH-001556 Mero-25  
## 1307 ACH-001557 Mero-41  
## 1308 ACH-001558 Mero-48a  
## 1309 ACH-001559 Mero-82  
## 1310 ACH-001560 Mero-83  
## 1311 ACH-001561 Mero-84  
## 1312 ACH-001562 Mero-95  
## 1313 ACH-001563 MM127  
## 1314 ACH-001566 MM370  
## 1315 ACH-001567 MM383  
## 1316 ACH-001568 MM386  
## 1317 ACH-001569 MM415  
## 1318 ACH-001570 MM426  
## 1319 ACH-001573 MOLM-1  
## 1320 ACH-001574 MOLM-14  
## 1321 ACH-001577 MUTZ-8  
## 1322 ACH-001578 NCCIT  
## 1323 ACH-001591 NCI-H1417  
## 1324 ACH-001599 NCI-H64  
## 1325 ACH-001603 NH-12  
## 1326 ACH-001605 no.10  
## 1327 ACH-001606 no.11  
## 1328 ACH-001607 NOZ  
## 1329 ACH-001608 NP 2  
## 1330 ACH-001609 NP 3  
## 1331 ACH-001610 NP 5  
## 1332 ACH-001611 NP 8  
## 1333 ACH-001613 OCI-AML4  
## 1334 ACH-001616 OCI-LY18  
## 1335 ACH-001617 OCI-LY7  
## 1336 ACH-001618 OCI-M2  
## 1337 ACH-001619 OCUG-1  
## 1338 ACH-001622 Onda 7  
## 1339 ACH-001623 Onda 8  
## 1340 ACH-001624 Onda 9  
## 1341 ACH-001625 OSC-19  
## 1342 ACH-001626 OSC-20  
## 1343 ACH-001627 P4E6  
## 1344 ACH-001628 PEA1  
## 1345 ACH-001630 PEO1  
## 1346 ACH-001632 PEO4  
## 1347 ACH-001634 PGA-1  
## 1348 ACH-001636 Ramos  
## 1349 ACH-001638 RC-K8  
## 1350 ACH-001639 ROS-50  
## 1351 ACH-001641 SAT  
## 1352 ACH-001642 SCC-3  
## 1353 ACH-001645 SEKI  
## 1354 ACH-001647 SHI-1  
## 1355 ACH-001648 Shmac 4  
## 1356 ACH-001649 Shmac 5  
## 1357 ACH-001650 SISO  
## 1358 ACH-001651 SKG-I  
## 1359 ACH-001652 SKG-II  
## 1360 ACH-001653 SK-GT-2  
## 1361 ACH-001654 SK-GT-4  
## 1362 ACH-001655 SKN  
## 1363 ACH-001656 SKNO-1  
## 1364 ACH-001664 SNU-638  
## 1365 ACH-001668 SUSA  
## 1366 ACH-001669 TANOUE  
## 1367 ACH-001670 TASK1  
## 1368 ACH-001673 TFK-1  
## 1369 ACH-001674 TGW  
## 1370 ACH-001675 TR146  
## 1371 ACH-001677 U-2904  
## 1372 ACH-001680 U-698-M  
## 1373 ACH-001685 U-HO1  
## 1374 ACH-001687 UM-RC-3  
## 1375 ACH-001688 UM-RC-7  
## 1376 ACH-001690 UPCI-SCC-026  
## 1377 ACH-001691 UPCI-SCC-029A  
## 1378 ACH-001692 UPCI-SCC-040  
## 1379 ACH-001694 UPCI-SCC-074  
## 1380 ACH-001696 UPCI-SCC-111  
## 1381 ACH-001698 UPCI-SCC-116  
## 1382 ACH-001699 UPCI-SCC-131  
## 1383 ACH-001701 UPCI-SCC-200  
## 1384 ACH-001702 VA-ES-BJ  
## 1385 ACH-001703 VAL  
## 1386 ACH-001704 VMRC-MELG  
## 1387 ACH-001707 WA-OSEL  
## 1388 ACH-001709 WSU-NHL  
## 1389 ACH-001711 PFSK-1  
## 1390 ACH-001712 Hs 860.T  
## 1391 ACH-001715 CAL-72  
## 1392 ACH-001716 GOTO  
## 1393 ACH-001719 OCI-C4P  
## 1394 ACH-001735 SEMK2  
## 1395 ACH-001736 HB1119  
## 1396 ACH-001737 CTV-1-DM  
## 1397 ACH-001738 CCRF-CEM  
## 1398 ACH-001740 RH28  
## 1399 ACH-001743 RC2  
## 1400 ACH-001745 RhJT  
## 1401 ACH-001750 TTC442  
## 1402 ACH-001751 Rh36  
## 1403 ACH-001765 Rh4  
## 1404 ACH-001767 CCLF\_PEDS\_0018\_T  
## 1405 ACH-001786 SNU-1544  
## 1406 ACH-001791 LPS6  
## 1407 ACH-001793 LPS27  
## 1408 ACH-001794 93T449  
## 1409 ACH-001795 94T778  
## 1410 ACH-001796 95T1000  
## 1411 ACH-001799 LPS141  
## 1412 ACH-001802 LPS853  
## 1413 ACH-001804 LPS510  
## 1414 ACH-001807 LPS067  
## 1415 ACH-001814 OS252  
## 1416 ACH-001818 C396  
## 1417 ACH-001819 MFM-223  
## 1418 ACH-001820 COLO 824  
## 1419 ACH-001825 SW527  
## 1420 ACH-001827 184B5  
## 1421 ACH-001834 ICC10  
## 1422 ACH-001835 ICC10-6  
## 1423 ACH-001836 ICC10-8  
## 1424 ACH-001838 ICC12  
## 1425 ACH-001839 ICC13-7  
## 1426 ACH-001841 ICC15  
## 1427 ACH-001842 ICC2  
## 1428 ACH-001843 ICC3  
## 1429 ACH-001844 ICC4  
## 1430 ACH-001845 ICC5  
## 1431 ACH-001846 ICC6  
## 1432 ACH-001848 ICC8  
## 1433 ACH-001849 ICC9  
## 1434 ACH-001850 G415  
## 1435 ACH-001852 HKGZ-CC  
## 1436 ACH-001853 KMCH-1  
## 1437 ACH-001856 RBE  
## 1438 ACH-001857 SG231  
## 1439 ACH-001858 SSP-25  
## 1440 ACH-001861 TGBC1TKB  
## 1441 ACH-001862 TGBC52TKB  
## 1442 ACH-001863 TKKK  
## 1443 ACH-001864 YSCCC  
## 1444 ACH-001958 MS1  
## 1445 ACH-001959 CC-LP-1  
## 1446 ACH-001960 CC-SW-1  
## 1447 ACH-001961 GB2  
## 1448 ACH-001970 MM253  
## 1449 ACH-001973 MM485  
## 1450 ACH-001977 NO36  
## 1451 ACH-001982 NZM3  
## 1452 ACH-001986 NZM42  
## 1453 ACH-001990 NZM7  
## 1454 ACH-001991 NZOV9  
## 1455 ACH-001992 ONE58  
## 1456 ACH-001993 NALM-16  
## 1457 ACH-001997 ECC2  
## 1458 ACH-001999 950-5-BIK  
## 1459 ACH-002001 A375 SKIN CJ1  
## 1460 ACH-002002 A375 SKIN CJ2  
## 1461 ACH-002003 A375 SKIN CJ3  
## 1462 ACH-002004 UACC62 SKIN CJ1  
## 1463 ACH-002005 SK-MEL-19  
## 1464 ACH-002011 MP46  
## 1465 ACH-002014 Mel270  
## 1466 ACH-002015 Mel285  
## 1467 ACH-002016 Mel290  
## 1468 ACH-002017 Omm1  
## 1469 ACH-002018 Omm2.5  
## 1470 ACH-002019 HOKUG  
## 1471 ACH-002020 SKG-IIIa  
## 1472 ACH-002021 T3M-3  
## 1473 ACH-002023 TGBC18TKB  
## 1474 ACH-002024 ECC4  
## 1475 ACH-002025 TT1TKB  
## 1476 ACH-002026 HHUA  
## 1477 ACH-002027 HOUA-I  
## 1478 ACH-002029 SAS  
## 1479 ACH-002035 LCAM1  
## 1480 ACH-002038 HSKTC  
## 1481 ACH-002039 PK-8  
## 1482 ACH-002040 HMV-II  
## 1483 ACH-002041 HOTHC  
## 1484 ACH-002042 T3M-5  
## 1485 ACH-002043 Ca9-22  
## 1486 ACH-002044 HSQ-89  
## 1487 ACH-002045 HO-1-u-1  
## 1488 ACH-002046 HTMMT  
## 1489 ACH-002048 RMS-YM  
## 1490 ACH-002051 Lu-134-A  
## 1491 ACH-002052 Lu-139  
## 1492 ACH-002055 TL-1  
## 1493 ACH-002058 ATN-1  
## 1494 ACH-002059 P30/OHK  
## 1495 ACH-002061 P2UR/K-562  
## 1496 ACH-002062 SLVL  
## 1497 ACH-002065 HS-PSS  
## 1498 ACH-002066 HS-Sch-2  
## 1499 ACH-002067 NOS-1  
## 1500 ACH-002069 HS-Os-1  
## 1501 ACH-002070 HKBMM  
## 1502 ACH-002077 Lu-165  
## 1503 ACH-002080 TN-2  
## 1504 ACH-002083 NB69  
## 1505 ACH-002084 MMAc  
## 1506 ACH-002085 HS-SY-II  
## 1507 ACH-002089 201T  
## 1508 ACH-002090 BB65-RCC  
## 1509 ACH-002091 CAL-39  
## 1510 ACH-002092 CHSA0011  
## 1511 ACH-002093 CHSA0108  
## 1512 ACH-002094 CHSA8926  
## 1513 ACH-002095 COR-L303  
## 1514 ACH-002096 CP50-MEL-B  
## 1515 ACH-002097 CP66-MEL  
## 1516 ACH-002098 CP67-MEL  
## 1517 ACH-002099 CS1  
## 1518 ACH-002100 DJM-1  
## 1519 ACH-002101 EMC-BAC-1  
## 1520 ACH-002102 EMC-BAC-2  
## 1521 ACH-002103 ES1  
## 1522 ACH-002104 ES3  
## 1523 ACH-002105 ES4  
## 1524 ACH-002106 ES5  
## 1525 ACH-002107 ES6  
## 1526 ACH-002108 ES7  
## 1527 ACH-002109 ES8  
## 1528 ACH-002110 EW-1  
## 1529 ACH-002111 EW-11  
## 1530 ACH-002112 EW-12  
## 1531 ACH-002113 EW-13  
## 1532 ACH-002114 EW-16  
## 1533 ACH-002115 EW-18  
## 1534 ACH-002116 EW-22  
## 1535 ACH-002117 EW-24  
## 1536 ACH-002118 EW-3  
## 1537 ACH-002119 EW-7  
## 1538 ACH-002120 GAK  
## 1539 ACH-002121 G-MEL  
## 1540 ACH-002122 GT3TKB  
## 1541 ACH-002123 H2369  
## 1542 ACH-002124 H2373  
## 1543 ACH-002125 H2461  
## 1544 ACH-002126 H2591  
## 1545 ACH-002127 H2595  
## 1546 ACH-002128 H2722  
## 1547 ACH-002129 H2731  
## 1548 ACH-002130 H2795  
## 1549 ACH-002131 H2803  
## 1550 ACH-002132 H2804  
## 1551 ACH-002133 H2810  
## 1552 ACH-002134 H2818  
## 1553 ACH-002135 H2869  
## 1554 ACH-002136 H290  
## 1555 ACH-002138 H513  
## 1556 ACH-002139 HA7-RCC  
## 1557 ACH-002140 Hey  
## 1558 ACH-002141 HSC-39  
## 1559 ACH-002142 HuO-3N1  
## 1560 ACH-002143 IST-MEL1  
## 1561 ACH-002144 IST-SL1  
## 1562 ACH-002145 IST-SL2  
## 1563 ACH-002146 JHOS-3  
## 1564 ACH-002147 K2  
## 1565 ACH-002148 K5  
## 1566 ACH-002149 KGN  
## 1567 ACH-002150 LB1047-RCC  
## 1568 ACH-002151 LB2241-RCC  
## 1569 ACH-002152 LB2518-MEL  
## 1570 ACH-002153 LB373-MEL-D  
## 1571 ACH-002154 LB647-SCLC  
## 1572 ACH-002155 LB996-RCC  
## 1573 ACH-002156 LC-1-sq  
## 1574 ACH-002157 LC-2-ad  
## 1575 ACH-002158 LU-99A  
## 1576 ACH-002159 M14  
## 1577 ACH-002160 MC-IXC  
## 1578 ACH-002161 MKN28  
## 1579 ACH-002162 MMAC-SF  
## 1580 ACH-002163 MRK-nu-1  
## 1581 ACH-002164 MZ1-PC  
## 1582 ACH-002165 MZ2-MEL  
## 1583 ACH-002166 MZ7-mel  
## 1584 ACH-002167 NCC010  
## 1585 ACH-002168 NCC021  
## 1586 ACH-002169 NCI-H1304  
## 1587 ACH-002170 NCI-H1688  
## 1588 ACH-002171 NCI-H250  
## 1589 ACH-002172 NCI-H322M  
## 1590 ACH-002173 NCI-H378  
## 1591 ACH-002174 NCI-H720  
## 1592 ACH-002175 NCI-H740  
## 1593 ACH-002177 NCI-H835  
## 1594 ACH-002178 NY  
## 1595 ACH-002179 OCUB-M  
## 1596 ACH-002180 OMC-1  
## 1597 ACH-002181 OVCA420  
## 1598 ACH-002182 OVCA433  
## 1599 ACH-002183 OVMIU  
## 1600 ACH-002184 PC-3\_[JPC-3]  
## 1601 ACH-002185 PL18  
## 1602 ACH-002186 PL4  
## 1603 ACH-002187 RCC-AB  
## 1604 ACH-002188 RCC-ER  
## 1605 ACH-002189 RCC-FG2  
## 1606 ACH-002190 RCC-JF  
## 1607 ACH-002191 RCC-JW  
## 1608 ACH-002192 RCC-MF  
## 1609 ACH-002193 RERF-LC-FM  
## 1610 ACH-002195 RXF393  
## 1611 ACH-002196 SBC-1  
## 1612 ACH-002197 SBC-3  
## 1613 ACH-002198 SCH  
## 1614 ACH-002199 SN12C  
## 1615 ACH-002200 SW962  
## 1616 ACH-002201 TC-YIK  
## 1617 ACH-002202 TMK-1  
## 1618 ACH-002204 U-CH2  
## 1619 ACH-002205 WM1552C  
## 1620 ACH-002206 WM278  
## 1621 ACH-002207 WM35  
## 1622 ACH-002208 YMB-1-E  
## 1623 ACH-002209 ALL-PO  
## 1624 ACH-002210 ARH-77  
## 1625 ACH-002211 BALL-1  
## 1626 ACH-002212 BB30-HNC  
## 1627 ACH-002213 BB49-HNC  
## 1628 ACH-002214 BC-1  
## 1629 ACH-002215 BC-3  
## 1630 ACH-002216 BE-13  
## 1631 ACH-002217 BE2-M17  
## 1632 ACH-002218 CESS  
## 1633 ACH-002219 COLO-320-HSR  
## 1634 ACH-002220 CRO-AP2  
## 1635 ACH-002221 CTB-1  
## 1636 ACH-002222 CTV-1  
## 1637 ACH-002223 D-245MG  
## 1638 ACH-002224 D-247MG  
## 1639 ACH-002225 D-263MG  
## 1640 ACH-002226 D-336MG  
## 1641 ACH-002227 D-392MG  
## 1642 ACH-002228 D-423MG  
## 1643 ACH-002229 D-502MG  
## 1644 ACH-002230 D-542MG  
## 1645 ACH-002231 D-566MG  
## 1646 ACH-002232 DG-75  
## 1647 ACH-002233 DIFI  
## 1648 ACH-002234 DOK  
## 1649 ACH-002235 DSH1  
## 1650 ACH-002236 EB-3  
## 1651 ACH-002237 ETK-1  
## 1652 ACH-002238 GR-ST  
## 1653 ACH-002239 H3118  
## 1654 ACH-002240 H9  
## 1655 ACH-002241 HAL-01  
## 1656 ACH-002242 HC-1  
## 1657 ACH-002243 HCE-4  
## 1658 ACH-002244 HO-1-N-1  
## 1659 ACH-002245 Hs-445  
## 1660 ACH-002246 Hs 633.T  
## 1661 ACH-002247 IM-9  
## 1662 ACH-002248 IMR-5  
## 1663 ACH-002249 JHU-011  
## 1664 ACH-002250 JHU-022  
## 1665 ACH-002251 JHU-029  
## 1666 ACH-002252 JiyoyeP-2003  
## 1667 ACH-002253 JSC-1  
## 1668 ACH-002254 KARPAS-1106P  
## 1669 ACH-002255 KARPAS-231  
## 1670 ACH-002256 KARPAS-45  
## 1671 ACH-002257 KINGS-1  
## 1672 ACH-002258 KMOE-2  
## 1673 ACH-002259 KNS-81-FD  
## 1674 ACH-002261 KP-N-YS  
## 1675 ACH-002262 KY821  
## 1676 ACH-002263 KYSE-220  
## 1677 ACH-002264 KYSE-50  
## 1678 ACH-002265 LB771-HNC  
## 1679 ACH-002266 LB831-BLC  
## 1680 ACH-002267 LC4-1  
## 1681 ACH-002268 LN-405  
## 1682 ACH-002269 LNZTA3WT4  
## 1683 ACH-002270 MC-CAR  
## 1684 ACH-002271 MFH-ino  
## 1685 ACH-002272 MHH-PREB-1  
## 1686 ACH-002273 ML-2  
## 1687 ACH-002274 MLMA  
## 1688 ACH-002275 MN-60  
## 1689 ACH-002276 MY-M12  
## 1690 ACH-002277 NB(TU)1-10  
## 1691 ACH-002278 NB10  
## 1692 ACH-002279 NB12  
## 1693 ACH-002280 NB13  
## 1694 ACH-002281 NB14  
## 1695 ACH-002282 NB17  
## 1696 ACH-002283 NB5  
## 1697 ACH-002284 NB6  
## 1698 ACH-002285 NB7  
## 1699 ACH-002286 NCI-H128  
## 1700 ACH-002287 NCI-H630  
## 1701 ACH-002288 NEC8  
## 1702 ACH-002289 NK-92MI  
## 1703 ACH-002290 NKM-1  
## 1704 ACH-002291 NTERA-2 cl.D1  
## 1705 ACH-002292 OACp4C  
## 1706 ACH-002293 P32-ISH  
## 1707 ACH-002294 PCI-15A  
## 1708 ACH-002295 PCI-30  
## 1709 ACH-002296 PCI-38  
## 1710 ACH-002297 PCI-4B  
## 1711 ACH-002298 PCI-6A  
## 1712 ACH-002299 QIMR-WIL  
## 1713 ACH-002300 Ramos-2G6-4C10  
## 1714 ACH-002301 RF-48  
## 1715 ACH-002302 RPMI-8866  
## 1716 ACH-002304 SK-MG-1  
## 1717 ACH-002305 SKN-3  
## 1718 ACH-002306 STS-0421  
## 1719 ACH-002307 SU-DHL-16  
## 1720 ACH-002308 SUP-B8  
## 1721 ACH-002309 SW684  
## 1722 ACH-002310 SW872  
## 1723 ACH-002311 TE-12  
## 1724 ACH-002312 TGBC24TKB  
## 1725 ACH-002313 TK  
## 1726 ACH-002314 TUR  
## 1727 ACH-002316 WIL2-NS  
## 1728 ACH-002317 YT  
## 1729 ACH-002319 184A1  
## 1730 ACH-002320 600MPE  
## 1731 ACH-002321 HBL100  
## 1732 ACH-002322 HCC2185  
## 1733 ACH-002323 HCC2688  
## 1734 ACH-002324 HCC3153  
## 1735 ACH-002325 LY2  
## 1736 ACH-002326 MACLS2  
## 1737 ACH-002327 MCF12A  
## 1738 ACH-002328 MX1  
## 1739 ACH-002329 SKBR5  
## 1740 ACH-002330 SKBR7  
## 1741 ACH-002331 ZR75B  
## 1742 ACH-002332 GISTT1  
## 1743 ACH-002333 HCC827GR  
## 1744 ACH-002334 SS1A  
## 1745 ACH-002336 HCET  
## 1746 ACH-002337 JHU028  
## 1747 ACH-002338 M980513  
## 1748 ACH-002339 MOT  
## 1749 ACH-002340 NBSUSSR  
## 1750 ACH-002342 BB30PBL  
## 1751 ACH-002343 BB49EBV  
## 1752 ACH-002344 BB65EBV  
## 1753 ACH-002345 CaR-1  
## 1754 ACH-002346 CP50EBV  
## 1755 ACH-002347 CP66EBV  
## 1756 ACH-002348 HSJD-DIPG-007  
## 1757 ACH-002349 GBM001  
## 1758 ACH-002350 HA7EBV  
## 1759 ACH-002351 L542  
## 1760 ACH-002352 LB1047EBV  
## 1761 ACH-002353 LB2241EBV  
## 1762 ACH-002354 LB2518EBV  
## 1763 ACH-002355 LB373EBV  
## 1764 ACH-002356 LB647PBL  
## 1765 ACH-002357 LB771PBL  
## 1766 ACH-002358 LB831EBV  
## 1767 ACH-002360 MZ1B  
## 1768 ACH-002361 MZ7B  
## 1769 ACH-002362 NCIBL128  
## 1770 ACH-002363 NCIBL1395  
## 1771 ACH-002364 NCIBL1437  
## 1772 ACH-002365 NCIBL1770  
## 1773 ACH-002366 NCIBL2009  
## 1774 ACH-002367 NCIBL2052  
## 1775 ACH-002368 NCIBL2087  
## 1776 ACH-002369 NCIBL209  
## 1777 ACH-002370 NCIBL2122  
## 1778 ACH-002371 NCIBL2126  
## 1779 ACH-002372 NCIBL2171  
## 1780 ACH-002373 HCC1187BL  
## 1781 ACH-002374 HCC1599BL  
## 1782 ACH-002375 HCC1937BL  
## 1783 ACH-002376 LS1034-PBL  
## 1784 ACH-002377 HCC38BL  
## 1785 ACH-002378 HCC1143-BL  
## 1786 ACH-002379 J82EBV  
## 1787 ACH-002380 COLO829BL  
## 1788 ACH-002381 HCC2157BL  
## 1789 ACH-002382 HCC1395BL  
## 1790 ACH-002383 HCC2218BL  
## 1791 ACH-002384 HCC1954BL  
## 1792 ACH-002386 M00921  
## 1793 ACH-002387 M1203273  
## 1794 ACH-002388 MET2B  
## 1795 ACH-002389 ACN  
## 1796 ACH-002392 SC-1  
## 1797 ACH-002397 KMH-2  
## 1798 ACH-002399 21NT  
## 1799 ACH-002446 CCLF\_UPGI\_0005\_T  
## 1800 ACH-002458 HT144 SKIN FV1  
## 1801 ACH-002459 HT144 SKIN FV3  
## 1802 ACH-002460 HT144 SKIN FV2  
## 1803 ACH-002461 RVH421 SKIN FV1  
## 1804 ACH-002462 RPE1-ss48  
## 1805 ACH-002463 RPE1-ss77  
## 1806 ACH-002464 RPE1-ss6  
## 1807 ACH-002465 RPE1-ss119  
## 1808 ACH-002466 RPE1-ss111  
## 1809 ACH-002467 RPE1-ss51  
## 1810 ACH-002471 PSS008  
## 1811 ACH-002475 HAP1  
## 1812 ACH-002485 MAVER-1  
## 1813 ACH-002486 MES-OV  
## 1814 ACH-002508 WM3211  
## 1815 ACH-002509 WM4235  
## 1816 ACH-002510 M040416  
## 1817 ACH-002511 M140325  
## 1818 ACH-002512 MM160113  
## 1819 ACH-002523 SNU-739  
## 1820 ACH-002526 SNU-1327  
## 1821 ACH-002531 SNU-2535  
## 1822 ACH-002647 CCC-5  
## 1823 ACH-002650 ETCC-016  
## 1824 ACH-002654 JVE-015  
## 1825 ACH-002659 JVE-127  
## 1826 ACH-002664 JVE-253  
## 1827 ACH-002669 KP-363T  
## 1828 ACH-002672 MAPAC-HS-77  
## 1829 ACH-002680 170-MG-BA  
## 1830 ACH-002687 WM3772F  
## 1831 ACH-002693 S462  
## 1832 ACH-002710 MPNST-724  
## 1833 ACH-002730 CCLF\_UPGI\_0009\_T  
## 1834 ACH-002733 CCLF\_UPGI\_0040\_T  
## 1835 ACH-002736 CCLF\_UPGI\_0052\_T  
## 1836 ACH-002738 PANFR0420  
## 1837 ACH-002742 PANFR0233  
## 1838 ACH-002743 PANFR0368  
## 1839 ACH-002744 PANFR0402  
## 1840 ACH-002745 PANFR0069  
## 1841 ACH-002757 CCLF\_UPGI\_0068\_T  
## 1842 ACH-002785 NCC-LMS1-C1  
## 1843 ACH-002799 NCC-MPNST1-C1  
## 1844 ACH-002800 NCC-MPNST2-C1  
## 1845 ACH-002834 PSS131R  
## 1846 ACH-002835 CCLF\_UPGI\_0011\_T  
## 1847 ACH-002837 CCLF\_UPGI\_0015\_T  
## 1848 ACH-002839 CCLF\_CORE\_0002\_T  
## 1849 ACH-002847 YUHOIN 06-50  
## 1850 ACH-002871 CCLF\_UPGI\_0054\_T  
## 1851 ACH-002883 IPM-BO-053  
## 1852 ACH-002884 IPM-BO-055  
## 1853 ACH-002885 IPM-BO-056  
## 1854 ACH-002922 SK-N-MM  
## 1855 ACH-002926 UPMD1  
## 1856 ACH-002950 NH93T  
## 1857 ACH-002951 NH84T  
## 1858 ACH-002954 CCLF\_UPGI\_0012\_T  
## 1859 ACH-002967 CCLF\_UPGI\_0036\_T  
## 1860 ACH-002968 CCLF\_UPGI\_0041\_T  
## 1861 ACH-002972 CCLF\_UPGI\_0085\_T  
## 1862 ACH-002979 CCLF\_UPGI\_0101\_T  
## 1863 ACH-002981 CCLF\_UPGI\_0027\_T  
## 1864 ACH-003071 NCI-H748

The bracket operation on a dataframe can be difficult to interpret because multiple expression for the row and column indicies is a lot of information for one line of code. You will see easier-to-read functions for dataframe subsetting in the next lesson.

Lastly, try running View(metadata) in RStudio Console…whew, a nice way to examine your dataframe like a spreadsheet program!

# 4 Data Wrangling with Tidy Data, Part 1

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.3

## Warning: package 'purrr' was built under R version 4.0.5

## Warning: package 'stringr' was built under R version 4.0.3

load(url("https://github.com/fhdsl/Intro\_to\_R/raw/main/classroom\_data/CCLE.RData"))

## 4.1 Data Science Workflow



We are now equipped with enough fundamental programming skills to apply it to various steps in the data science workflow. We start with *Transform* and *Visualize* with the assumption that our data is in a nice, “Tidy format”. First, we need to understand what it means for a data to be “Tidy”.

## 4.2 Tidy Data

Here, we describe a standard of organizing data. It is important to have standards, as it facilitates a consistent way of thinking about data organization and building tools (functions) that make use of that standard. The principles of **tidy data**, developed by Hadley Wickham:

1. Each variable must have its own column.
2. Each observation must have its own row.
3. Each value must have its own cell.

If you want to be technical about what variables and observations are, Hadley Wickham describes:

A **variable** contains all values that measure the same underlying attribute (like height, temperature, duration) across units. An **observation** contains all values measured on the same unit (like a person, or a day, or a race) across attributes.



Figure : A tidy dataframe.

## 4.3 Examples and counter-examples of Tidy Data:

Consider the following three datasets, which all contain the exact same information:

table1

## # A tibble: 6 × 4  
## country year cases population  
## <chr> <int> <int> <int>  
## 1 Afghanistan 1999 745 19987071  
## 2 Afghanistan 2000 2666 20595360  
## 3 Brazil 1999 37737 172006362  
## 4 Brazil 2000 80488 174504898  
## 5 China 1999 212258 1272915272  
## 6 China 2000 213766 1280428583

This table1 satisfies the the definition of Tidy Data. The observation is a country’s year, and the variables are attributes of each country’s year.

head(table2)

## # A tibble: 6 × 4  
## country year type count  
## <chr> <int> <chr> <int>  
## 1 Afghanistan 1999 cases 745  
## 2 Afghanistan 1999 population 19987071  
## 3 Afghanistan 2000 cases 2666  
## 4 Afghanistan 2000 population 20595360  
## 5 Brazil 1999 cases 37737  
## 6 Brazil 1999 population 172006362

Something is strange able table2. The observation is still a country’s year, but “type” and “count” are not clear attributes of each country’s year.

table3

## # A tibble: 6 × 3  
## country year rate   
## \* <chr> <int> <chr>   
## 1 Afghanistan 1999 745/19987071   
## 2 Afghanistan 2000 2666/20595360   
## 3 Brazil 1999 37737/172006362   
## 4 Brazil 2000 80488/174504898   
## 5 China 1999 212258/1272915272  
## 6 China 2000 213766/1280428583

In table3, we have multiple values for each cell under the “rate” column.

## 4.4 Our working Tidy Data: DepMap Project

The [Dependency Map project](https://depmap.org/portal/) is a multi-omics profiling of cancer cell lines combined with functional assays such as CRISPR and drug sensitivity to help identify cancer vulnerabilities and drug targets. Here are some of the data that we have public access. We have been looking at the metadata since last session.

* Metadata
* Somatic mutations
* Gene expression
* Drug sensitivity
* CRISPR knockout
* and more…

Let’s see how these datasets fit the definition of Tidy data:

| Dataframe | The observation is | Some variables are | Some values are |
| --- | --- | --- | --- |
| metadata | Cell line | ModelID, Age, OncotreeLineage | “ACH-000001”, 60, “Myeloid” |
| expression |  |  |  |
| mutation |  |  |  |

## 4.5 Transform: “What do you want to do with this dataframe”?

Remember that a major theme of the course is about: **How we organize ideas <-> Instructing a computer to do something.**

Until now, we haven’t focused too much on how we organize our scientific ideas to interact with what we can do with code. Let’s pivot to write our code driven by our scientific curiosity. After we are sure that we are working with Tidy data, we can ponder how we want to transform our data that satisfies our scientific question. We will look at several ways we can transform tidy data, starting with subsetting columns and rows.

Here’s a starting prompt:

In the metadata dataframe, which rows would you filter for and columns would you select that relate to a scientific question?

We should use the implicit subsetting mindset here: ie. “I want to filter for rows such that the Subtype is breast cancer and look at the Age and Sex.” and *not* “I want to filter for rows 20-50 and select columns 2 and 8”.

*Notice that when we filter for rows in an implicit way, we often formulate our criteria about the columns.*

(This is because we are guaranteed to have column names in dataframes, but not usually row names. Some dataframes have row names, but because the data types are not guaranteed to have the same data type across rows, it makes describing by row properties difficult.)

Let’s convert our implicit subsetting criteria into code!

metadata\_filtered = filter(metadata, OncotreeLineage == "Breast")  
breast\_metadata = select(metadata\_filtered, ModelID, Age, Sex)  
  
head(breast\_metadata)

## ModelID Age Sex  
## 1 ACH-000017 43 Female  
## 2 ACH-000019 69 Female  
## 3 ACH-000028 69 Female  
## 4 ACH-000044 47 Female  
## 5 ACH-000097 63 Female  
## 6 ACH-000111 41 Female

Here, filter() and select() are functions from the tidyverse package, which we have to install and load in via library(tidyverse) before using these functions.

### 4.5.1 Filter rows

Let’s carefully a look what how the R Console is interpreting the filter() function:

* We evaluate the expression right of =.
* The first argument of filter() is a dataframe, which we give metadata.
* The second argument is strange: the expression we give it looks like a logical indexing vector built from a comparison operator, but the variable OncotreeLineage does not exist in our environment! Rather, OncotreeLineage is a column from metadata, and we are referring to it as a **data variable** in the context of the dataframe metadata. So, we make a comparison operation on the column OncotreeLineage from metadata and its resulting logical indexing vector is the input to the second argument.
  + How do we know when a variable being used is a variable from the environment, or a data variable from a dataframe? It’s not clear cut, but here’s a rule of thumb: most functions from the tidyverse package allows you to use data variables to refer to columns of a dataframe. We refer to documentation when we are not sure.
  + This encourages more *readable* code at the expense of consistency of referring to variables in the environment. The authors of this package [describes this trade-off](https://dplyr.tidyverse.org/articles/programming.html#data--and-env-variables).
* Putting it together, filter() takes in a dataframe, and an logical indexing vector described by data variables as arguments, and returns a data frame with rows that match condition described by the logical indexing vector.
* Store this in metadata\_filtered variable.

### 4.5.2 Select columns

Let’s carefully a look what how the R Console is interpreting the select() function:

* We evaluate the expression right of =.
* The first argument of filter() is a dataframe, which we give metadata.
* The second and third arguments are **data variables** referring the columns of metadata.
  + For certain functions like filter(), there is no limit on the number of arguments you provide. You can keep adding data variables to select for more column names.
* Putting it together, select() takes in a dataframe, and as many data variables you like to select columns, and returns a dataframe with the columns you described by data variables.
* Store this in breast\_metadata variable.

## 4.6 Summary Statistics

Now that your dataframe has be transformed based on your scientific question, you can start doing some analysis on it! A common data science task is to examine summary statistics of a dataset, which summarizes the observations of a variable in a numeric summary.

If the columns of interest are numeric, then you can try functions such as mean(), median(), mode(), or summary() to get summary statistics of the column. If the columns of interest is character or logical, then you can try the table() function.

All of these functions take in a vector as input and not a dataframe, so you have to access the column as a vector via the $ operation.

mean(breast\_metadata$Age, na.rm = TRUE)

## [1] 50.96104

table(breast\_metadata$Sex)

##   
## Female Unknown   
## 91 1

## 4.7 Pipes

Often, in data analysis, we want to transform our dataframe in multiple steps via different functions. This leads to nested function calls, like this:

breast\_metadata = select(filter(metadata, OncotreeLineage == "Breast"), ModelID, Age, Sex)

This is a bit hard to read. A computer doesn’t care how difficult it is to read this line of code, but there is a lot of instructions going on in one line of code. This multi-step function composition will lead to an unreadable pattern such as:

result = function3(function2(function1(dataframe, df\_col4, df\_col2), arg2), df\_col5, arg1)

To untangle this, you have to look into the middle of this code, and slowly step out of it.

To make this more readable, programmers came up with an alternative syntax for function composition via the **pipe** metaphor. The ideas is that we push data through a chain of connected pipes, in which the output of a pipe becomes the input of the subsequent pipe.

Instead of a syntax like result2 = function3(function2(function1(dataframe))),

we linearize it with the %>% symbol: result2 = dataframe %>% function1 %>% function2 %>% function3.

In the previous example,

result = dataframe %>% function1(df\_col4, df\_col2) %>%  
 function2(arg2) %>%  
 function3(df\_col5, arg1)

This looks much easier to read. Notice that we have broken up one expression in to three lines of code for readability. If a line of code is incomplete (the first line of code is piping to somewhere unfinished), the R will treat the next line of code as part of the current line of code.

Try to rewrite the select() and filter() function composition example above using the pipe metaphor and syntax.

# 5 Data Wrangling with Tidy Data, Part 2

Today, we will continue learning about common functions from the Tidyverse that is useful for Tidy data manipulations.

## 5.1 Modifying and creating new columns in dataframes

The mutate() function takes in the following arguments: the first argument is the dataframe of interest, and the second argument is a *new or existing data variable* that is defined in terms of *other data variables*.

We create a new column olderAge that is 10 years older than the original Age column.

metadata$Age[1:10]

## [1] 60 36 72 30 30 64 63 56 72 53

metadata2 = mutate(metadata, olderAge = Age + 10)  
metadata2$olderAge[1:10]

## [1] 70 46 82 40 40 74 73 66 82 63

Here, we used an operation on a column of metadata. Here’s another example with a function:

expression$KRAS\_Exp[1:10]

## [1] 4.634012 4.638653 4.032101 5.503031 3.713696 3.972693 3.235727 4.135042  
## [9] 9.017365 3.940167

expression2 = mutate(expression, log\_KRAS\_Exp = log(KRAS\_Exp))  
expression2$log\_KRAS\_Exp[1:10]

## [1] 1.533423 1.534424 1.394288 1.705299 1.312028 1.379444 1.174254 1.419498  
## [9] 2.199152 1.371223

### 5.1.1 Alternative: Creating and modifying columns via $

Instead of mutate() function, we can also create a new or modify a column via the $ symbol:

expression2 = expression  
expression2$log\_KRAS\_Exp = log(expression2$KRAS\_Exp)

## 5.2 Merging two dataframes together

Suppose we have the following dataframes:

expression

| ModelID | PIK3CA\_Exp | log\_PIK3CA\_Exp |
| --- | --- | --- |
| “ACH-001113” | 5.138733 | 1.636806 |
| “ACH-001289” | 3.184280 | 1.158226 |
| “ACH-001339” | 3.165108 | 1.152187 |

metadata

| ModelID | OncotreeLineage | Age |
| --- | --- | --- |
| “ACH-001113” | “Lung” | 69 |
| “ACH-001289” | “CNS/Brain” | NA |
| “ACH-001339” | “Skin” | 14 |

Suppose that I want to compare the relationship between OncotreeLineage and PIK3CA\_Exp, but they are columns in different dataframes. We want a new dataframe that looks like this:

| ModelID | PIK3CA\_Exp | log\_PIK3CA\_Exp | OncotreeLineage | Age |
| --- | --- | --- | --- | --- |
| “ACH-001113” | 5.138733 | 1.636806 | “Lung” | 69 |
| “ACH-001289” | 3.184280 | 1.158226 | “CNS/Brain” | NA |
| “ACH-001339” | 3.165108 | 1.152187 | “Skin” | 14 |

We see that in both dataframes, the rows (observations) represent cell lines with a common column ModelID, so let’s merge these two dataframes together, using full\_join():

merged = full\_join(metadata, expression, by = "ModelID")

The number of rows and columns of metadata:

dim(metadata)

## [1] 1864 30

The number of rows and columns of expression:

dim(expression)

## [1] 1450 536

The number of rows and columns of merged:

dim(merged)

## [1] 1864 565

We see that the number of *columns* in merged combines the number of columns in metadata and expression, while the number of *rows* in merged is the larger of the number of rows in metadata and expression : full\_join() keeps all observations common to both dataframes based on the common column defined via the by argument.

Therefore, we expect to see NA values in merged, as there are some cell lines that are not in expression dataframe.

There are variations of this function depending on your application:



Given xxx\_join(x, y, by = "common\_col"),

* full\_join() keeps all observations.
* left\_join() keeps all observations in x.
* right\_join() keeps all observations in y.
* inner\_join() keeps observations common to both x and y.

## 5.3 Grouping and summarizing dataframes

Also known as: “The rows I want is described by a column. The columns I want need to be summarized from other columns.”

In a dataset, there may be multiple levels of observations, and which level of observation we examine depends on our scientific question. For instance, in metadata, the observation is cell lines. However, perhaps we want to understand properties of metadata in which the observation is the cancer type, OncotreeLineage. Suppose we want the mean age of each cancer type, and the number of cell lines that we have for each cancer type.

This is a scenario in which the *desired rows are described by a column*, OncotreeLineage, and the columns, such as mean age, need to be *summarized from other columns.*

As an example, this dataframe is transformed from:

| ModelID | OncotreeLineage | Age |
| --- | --- | --- |
| “ACH-001113” | “Lung” | 69 |
| “ACH-001289” | “Lung” | 23 |
| “ACH-001339” | “Skin” | 14 |
| “ACH-002342” | “Brain” | 23 |
| “ACH-004854” | “Brain” | 56 |
| “ACH-002921” | “Brain” | 67 |

into:

| OncotreeLineage | MeanAge | Count |
| --- | --- | --- |
| “Lung” | 46 | 2 |
| “Skin” | 14 | 1 |
| “Brain” | 48.67 | 3 |

We use the functions group\_by() and summarise() :

metadata\_by\_type = metadata %>%   
 group\_by(OncotreeLineage) %>%   
 summarise(MeanAge = mean(Age, rm.na=TRUE), Count = n())

## `summarise()` ungrouping output (override with `.groups` argument)

Or, without pipes:

metadata\_by\_type\_temp = group\_by(metadata, OncotreeLineage)  
metadata\_by\_type = summarise(metadata\_by\_type\_temp, MeanAge = mean(Age, rm.na=TRUE), Count = n())

## `summarise()` ungrouping output (override with `.groups` argument)

The group\_by() function returns the identical input dataframe but remembers which variable(s) have been marked as grouped:

head(group\_by(metadata, OncotreeLineage))

## # A tibble: 6 × 30  
## # Groups: OncotreeLineage [3]  
## ModelID PatientID CellLineName StrippedCellLineName Age SourceType  
## <chr> <chr> <chr> <chr> <dbl> <chr>   
## 1 ACH-000001 PT-gj46wT NIH:OVCAR-3 NIHOVCAR3 60 Commercial  
## 2 ACH-000002 PT-5qa3uk HL-60 HL60 36 Commercial  
## 3 ACH-000003 PT-puKIyc CACO2 CACO2 72 Commercial  
## 4 ACH-000004 PT-q4K2cp HEL HEL 30 Commercial  
## 5 ACH-000005 PT-q4K2cp HEL 92.1.7 HEL9217 30 Commercial  
## 6 ACH-000006 PT-ej13Dz MONO-MAC-6 MONOMAC6 64 Commercial  
## # ℹ 24 more variables: SangerModelID <chr>, RRID <chr>, DepmapModelType <chr>,  
## # AgeCategory <chr>, GrowthPattern <chr>, LegacyMolecularSubtype <chr>,  
## # PrimaryOrMetastasis <chr>, SampleCollectionSite <chr>, Sex <chr>,  
## # SourceDetail <chr>, LegacySubSubtype <chr>, CatalogNumber <chr>,  
## # CCLEName <chr>, COSMICID <dbl>, PublicComments <chr>,  
## # WTSIMasterCellID <dbl>, EngineeredModel <chr>, TreatmentStatus <chr>,  
## # OnboardedMedia <chr>, PlateCoating <chr>, OncotreeCode <chr>, …

The summarise() returns one row for each combination of grouping variables, and one column for each of the summary statistics that you have specified.

Functions you can use for summarise() must take in a vector and return a simple data type, such as any of our summary statistics functions: mean(), median(), min(), max(), etc.

The exception is n(), which returns the number of entries for each grouping variable’s value.

You can combine group\_by() with other functions. See this [guide](https://dplyr.tidyverse.org/articles/grouping.html).

## 5.4 How functions are built

As you become more independent R programmers, you will spend time learning about new functions on your own. We have gone over the basic anatomy of a function call back in the first lesson, but now let’s go a bit deeper to understand how a function is built and how to call them.

Recall that a function has a **function name**, **input arguments**, and a **return value**.

*Function definition consists of assigning a* ***function name*** *with a “function” statement that has a comma-separated list of named* ***function arguments****, and a* ***return expression****. The function name is stored as a variable in the global environment.*

In order to use the function, one defines or import it, then one calls it.

Example:

addFunction = function(num1, num2) {  
 result = num1 + num2   
 return(result)  
}  
result = addFunction(3, 4)

With function definitions, not all code runs from top to bottom. The first four lines defines the function, but the function is never run. It is called on line 5, and the lines within the function are executed.

When the function is called in line 5, the variables for the arguments are reassigned to function arguments to be used within the function and helps with the modular form.

To see why we need the variables of the arguments to be reassigned, consider the following function that is *not* modular:

x = 3  
y = 4  
addFunction = function(num1, num2) {  
 result = x + y   
 return(result)  
}  
result = addFunction(10, -10)

Some syntax equivalents on calling the function:

addFunction(3, 4)  
addFunction(num1 = 3, num2 = 4)  
addFunction(num2 = 4, num1 = 3)

but this *could* be different:

addFunction(4, 3)

With a deeper knowledge of how functions are built, when you encounter a foreign function, you can look up its help page to understand how to use it. For example, let’s look at mean():

?mean  
  
Arithmetic Mean  
  
Description:  
  
 Generic function for the (trimmed) arithmetic mean.  
  
Usage:  
  
 mean(x, ...)  
   
 ## Default S3 method:  
 mean(x, trim = 0, na.rm = FALSE, ...)  
   
Arguments:  
  
 x: An R object. Currently there are methods for numeric/logical  
 vectors and date, date-time and time interval objects.  
 Complex vectors are allowed for ‘trim = 0’, only.  
  
 trim: the fraction (0 to 0.5) of observations to be trimmed from  
 each end of ‘x’ before the mean is computed. Values of trim  
 outside that range are taken as the nearest endpoint.  
  
 na.rm: a logical evaluating to ‘TRUE’ or ‘FALSE’ indicating whether  
 ‘NA’ values should be stripped before the computation  
 proceeds.  
  
 ...: further arguments passed to or from other methods.

Notice that the arguments trim = 0, na.rm = FALSE have default values. This means that these arguments are *optional* - you should provide it only if you want to. With this understanding, you can use mean() in a new way:

numbers = c(1, 2, NA, 4)  
mean(x = numbers, na.rm = TRUE)

## [1] 2.333333

# 6 Data Visualization

## 6.1 Common Plots

### 6.1.1 Univariate

* Numeric: histogram
* Character: bar plots

### 6.1.2 Bivariate

* Numeric vs. Numeric: Scatterplot, line plot
* Numeric vs. Character: Box plot

Why do we focus on these common plots? Our eyes are better at distinguishing certain visual features more than others. All of these plots are focused on their position to depict data, which gives us the most effective visual scale.



Figure : Source: <https://www.oreilly.com/library/view/visualization-analysis-and/9781466508910/K14708_C005.xhtml>

## 6.2 Grammar of Graphics

The syntax of the grammar of graphics breaks down into 4 sections.

Data

Mapping to data

Geometry

Additional settings

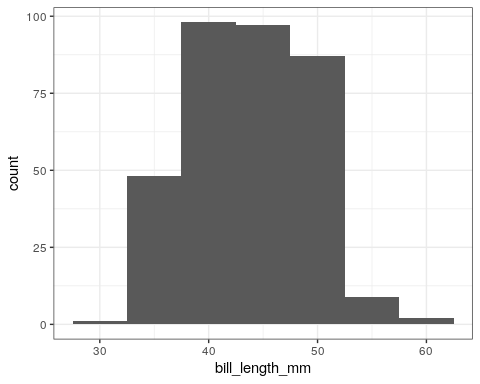
### 6.2.1 Histogram

ggplot(penguins) + aes(x = bill\_length\_mm) + geom\_histogram() + theme\_bw()



With options:

ggplot(penguins) + aes(x = bill\_length\_mm) + geom\_histogram(binwidth = 5) + theme\_bw()



### 6.2.2 Bar plots

ggplot(penguins) + aes(x = species) + geom\_bar()



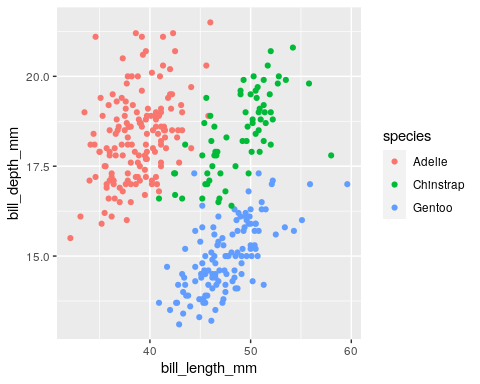
### 6.2.3 Scatterplot

ggplot(penguins) + aes(x = bill\_length\_mm, y = bill\_depth\_mm) + geom\_point()



### 6.2.4 Multivaraite Scatterplot

ggplot(penguins) + aes(x = bill\_length\_mm, y = bill\_depth\_mm, color = species) + geom\_point()



### 6.2.5 Multivaraite Scatterplot

ggplot(penguins) + aes(x = bill\_length\_mm, y = bill\_depth\_mm) + geom\_point() + facet\_wrap(~species)



### 6.2.6 Line plot?

ggplot(penguins) + aes(x = bill\_length\_mm, y = bill\_depth\_mm) + geom\_line()



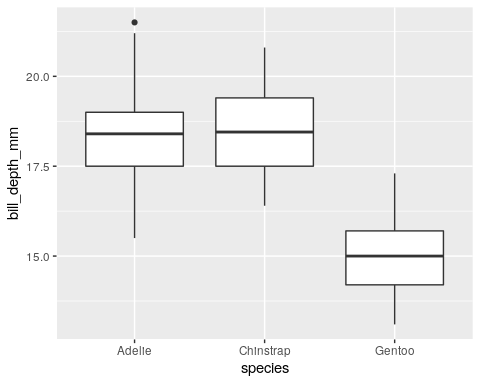
### 6.2.7 Grouped Line plot?

ggplot(penguins) + aes(x = bill\_length\_mm, y = bill\_depth\_mm, group = species) + geom\_line()



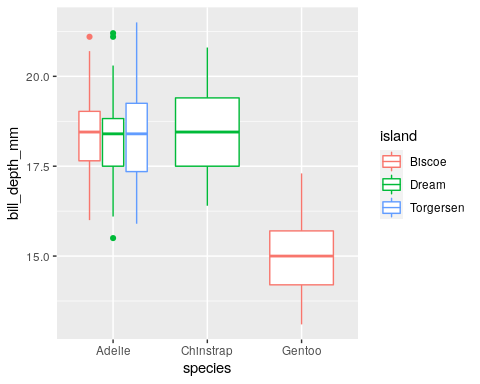
### 6.2.8 Boxplot

ggplot(penguins) + aes(x = species, y = bill\_depth\_mm) + geom\_boxplot()



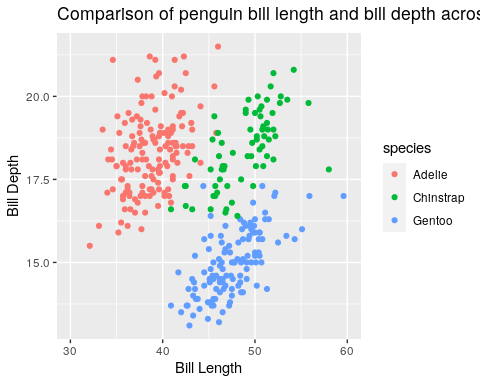
### 6.2.9 Grouped Boxplot

ggplot(penguins) + aes(x = species, y = bill\_depth\_mm, color = island) + geom\_boxplot()



### 6.2.10 Some additional options

ggplot(data = penguins) + aes(x = bill\_length\_mm, y = bill\_depth\_mm, color = species) + geom\_point() + labs(x = “Bill Length”, y = “Bill Depth”, title = “Comparison of penguin bill length and bill depth across species”) + scale\_x\_continuous(limits = c(30, 60))



## 6.3 Summary of options

data

geom\_point: x, y, color, shape

geom\_line: x, y, group, color

geom\_histogram: x, y, fill

geom\_bar: x, fill

geom\_boxplot: x, y, fill, color

facet\_wrap

labs

scale\_x\_continuous

scale\_y\_continuous

scale\_x\_discrete

scale\_y\_discrete

Consider the esquisse package to help generate your ggplot code via drag and drop.

# About the Authors

These credits are based on our [course contributors table guidelines](https://www.ottrproject.org/more_features.html#giving-credits-to-contributors).

| Credits | Names |
| --- | --- |
| **Pedagogy** |  |
| Lead Content Instructor(s) | Chris Lo |
| Lecturer | Chris Lo |
| Content Author(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | If any other authors besides lead instructor |
| Content Contributor(s) (include section name/link in parentheses) - make new line if more than one section involved | Wrote less than a chapter |
| Content Editor(s)/Reviewer(s) | Checked your content |
| Content Director(s) | Helped guide the content direction |
| Content Consultants (include chapter name/link in parentheses or word “General”) - make new line if more than one chapter involved | Gave high level advice on content |
| Acknowledgments | Gave small assistance to content but not to the level of consulting |
| **Production** |  |
| Content Publisher(s) | Helped with publishing platform |
| Content Publishing Reviewer(s) | Reviewed overall content and aesthetics on publishing platform |
| **Technical** |  |
| Course Publishing Engineer(s) | Helped with the code for the technical aspects related to the specific course generation |
| Template Publishing Engineers | [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/), [Ava Hoffman](https://www.avahoffman.com/) |
| Publishing Maintenance Engineer | [Candace Savonen](https://www.cansavvy.com/) |
| Technical Publishing Stylists | [Carrie Wright](https://carriewright11.github.io/), [Ava Hoffman](https://www.avahoffman.com/), [Candace Savonen](https://www.cansavvy.com/) |
| Package Developers ([ottrpal](https://github.com/jhudsl/ottrpal)) [Candace Savonen](https://www.cansavvy.com/), [John Muschelli](https://johnmuschelli.com/), [Carrie Wright](https://carriewright11.github.io/) |  |
| **Art and Design** |  |
| Illustrator(s) | Created graphics for the course |
| Figure Artist(s) | Created figures/plots for course |
| Videographer(s) | Filmed videos |
| Videography Editor(s) | Edited film |
| Audiographer(s) | Recorded audio |
| Audiography Editor(s) | Edited audio recordings |
| **Funding** |  |
| Funder(s) | Institution/individual who funded course including grant number |
| Funding Staff | Staff members who help with funding |

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value   
## version R version 4.0.2 (2020-06-22)  
## os Ubuntu 20.04.5 LTS   
## system x86\_64, linux-gnu   
## ui X11   
## language (EN)   
## collate en\_US.UTF-8   
## ctype en\_US.UTF-8   
## tz Etc/UTC   
## date 2024-01-06   
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date lib source   
## assertthat 0.2.1 2019-03-21 [1] RSPM (R 4.0.5)   
## bookdown 0.24 2023-03-28 [1] Github (rstudio/bookdown@88bc4ea)   
## cachem 1.0.7 2023-02-24 [1] CRAN (R 4.0.2)   
## callr 3.5.0 2020-10-08 [1] RSPM (R 4.0.2)   
## cli 3.6.1 2023-03-23 [1] CRAN (R 4.0.2)   
## crayon 1.3.4 2017-09-16 [1] RSPM (R 4.0.0)   
## desc 1.2.0 2018-05-01 [1] RSPM (R 4.0.3)   
## devtools 2.3.2 2020-09-18 [1] RSPM (R 4.0.3)   
## digest 0.6.25 2020-02-23 [1] RSPM (R 4.0.0)   
## ellipsis 0.3.1 2020-05-15 [1] RSPM (R 4.0.3)   
## evaluate 0.20 2023-01-17 [1] CRAN (R 4.0.2)   
## fastmap 1.1.1 2023-02-24 [1] CRAN (R 4.0.2)   
## fs 1.5.0 2020-07-31 [1] RSPM (R 4.0.3)   
## glue 1.4.2 2020-08-27 [1] RSPM (R 4.0.5)   
## htmltools 0.5.5 2023-03-23 [1] CRAN (R 4.0.2)   
## knitr 1.33 2023-03-28 [1] Github (yihui/knitr@a1052d1)   
## magrittr 2.0.3 2022-03-30 [1] CRAN (R 4.0.2)   
## memoise 2.0.1 2021-11-26 [1] CRAN (R 4.0.2)   
## pkgbuild 1.1.0 2020-07-13 [1] RSPM (R 4.0.2)   
## pkgload 1.1.0 2020-05-29 [1] RSPM (R 4.0.3)   
## prettyunits 1.1.1 2020-01-24 [1] RSPM (R 4.0.3)   
## processx 3.4.4 2020-09-03 [1] RSPM (R 4.0.2)   
## ps 1.4.0 2020-10-07 [1] RSPM (R 4.0.2)   
## R6 2.4.1 2019-11-12 [1] RSPM (R 4.0.0)   
## remotes 2.2.0 2020-07-21 [1] RSPM (R 4.0.3)   
## rlang 1.1.0 2023-03-14 [1] CRAN (R 4.0.2)   
## rmarkdown 2.10 2023-03-28 [1] Github (rstudio/rmarkdown@02d3c25)  
## rprojroot 2.0.3 2022-04-02 [1] CRAN (R 4.0.2)   
## sessioninfo 1.1.1 2018-11-05 [1] RSPM (R 4.0.3)   
## stringi 1.5.3 2020-09-09 [1] RSPM (R 4.0.3)   
## stringr 1.4.0 2019-02-10 [1] RSPM (R 4.0.3)   
## testthat 3.0.1 2023-03-28 [1] Github (R-lib/testthat@e99155a)   
## usethis 1.6.3 2020-09-17 [1] RSPM (R 4.0.2)   
## withr 2.3.0 2020-09-22 [1] RSPM (R 4.0.2)   
## xfun 0.26 2023-03-28 [1] Github (yihui/xfun@74c2a66)   
## yaml 2.2.1 2020-02-01 [1] RSPM (R 4.0.3)   
##   
## [1] /usr/local/lib/R/site-library  
## [2] /usr/local/lib/R/library

# 7 References