

Data Science – Data Prep with R – Quick Reference

PROFILE DATASET

Volume	<code>df2 <- df %>% summarize (count = n())</code>
--------	--

Velocity	<code>df2 <- df %>% group_by (date1) %>% summarize (count = n())</code>
----------	--

Attribute Selection	<code>df2 <- df %>% select (c ('attr1' , 'attr2'))</code>
---------------------	---

Incomplete Records	<code>df2 <- df %>% filter (!is.na (attr1))</code>
--------------------	--

VALIDATE ATTRIBUTES

Domain	<code>distinct(df,attr1)</code>
--------	---------------------------------

Missing Values	<code>df2 <- df %>% filter(is.na (attr1))</code>
----------------	--

Range	<code>summary (df)</code>
-------	---------------------------

Data Types	<code>str (df)</code>
------------	-----------------------

Outliers	<code>summary (df); hist (df\$attr1)</code>
----------	---

Distribution	<code>hist (df\$attr1)</code>
--------------	-------------------------------

STANDARDIZE ATTRIBUTES

Data Types	<code>mutate (attr1 = as.integer (attr1), attr2 = factor (attr2), date1 = as.Date (date1))</code>
------------	---

Patterns	<code>mutate (attr1 = if_else (attr1 == 'Street', 'St', attr1)</code>
----------	---

Formatting	<code>mutate (attr1 = toupper (attr1))</code>
------------	---

Scaling	<code>mutate (attr1=scale (attr1))</code>
---------	---

CLEAN ATTRIBUTES

Outliers (Quantitative)	<code>mutate (attr1 = if_else (attr1 > 1000 attr1 < 0, NA, attr1)</code>
-------------------------	--

Missing Values (At Random)	<code>mutate(attr1 = if_else (is.na(attr1), mean(attr1, na.rm=TRUE), attr1))</code>
----------------------------	---

Missing Values (Not at Random)	<code>mutate(attr1 = if_else (is.na (attr1), 1, attr1)</code>
--------------------------------	---

Incorrect Values	<code>mutate(attr1 = if_else (attr1 == 'bad', 'good', attr1))</code>
------------------	--

DERIVE ATTRIBUTES

Buckets/Binning	<code>mutate (attr1_bin = cut(x = attr1, breaks = c(0,50,100)))</code>
-----------------	--

Date Parts	<code>mutate (month = format(date1, format = "%m")</code>
------------	---

Date Difference	<code>mutate (elapsed_days = difftime (date1, date2, units = 'days')</code>
-----------------	---

Last Period	<code>mutate (last_year = as.numeric (format(date1, "%Y"))-1</code>
-------------	---

Dummy Encoding (One Hot)	<code>mutate (gender_male = if_else (attr1 == 'male', 1, 0)</code>
--------------------------	--

COMBINE DATASETS

Join Horizontally (Full Match)	<code>df3 <- inner_join (x=df1, y=df2, by='attr1')</code>
--------------------------------	--

Join Horizontally (Optional Match)	<code>df3 <- left_join (x=df1, y=df2, by='attr1')</code>
------------------------------------	---

Union Vertically (Deduplicate)	<code>df3 <- rbind (df1, df2) df4 <- df3 [match (unique (df3\$attr1), df3\$attr1),]</code>
--------------------------------	---

Union Vertically (No Deduplicate)	<code>df3 <- rbind (df1, df2)</code>
-----------------------------------	---

SPLIT DATASETS

Simple Filter	<code>df2 <- df %>% filter(attr1>5)</code>
---------------	---

Filter Based on Aggregation	<code>df2 <- df %>% filter(attr1 > mean(attr1))</code>
-----------------------------	---

Sampling (Random)	<code>set.seed (100) df2 <- sample_n (df, 1000)</code>
-------------------	---

Sampling (Non-Random)	<code>df2 <- df %>% filter (ntile (attr1, 4) == 4)</code>
-----------------------	---

CREATE INTERFACE

Python	<code>library (reticulate)</code>
--------	-----------------------------------

SQL	<code>library (dbi)</code>
-----	----------------------------

All items assume **dplyr** is loaded from tidyverse package.
df is a dataframe with attributes attr1, attr2, date1, date2.