

R Data Science Quick Reference Cheat Sheet

MOST USED R FUNCTIONS - QUICK LOOKUP

=== DATA LOADING ===

```
read_csv("file.csv")      # Load CSV
read.csv("file.csv")      # Base R
read_excel("file.xlsx")   # Excel
write_csv(df, "file.csv") # Save CSV
```

=== BASIC OPERATIONS ===

```
head(df)                  # First 6 rows
tail(df)                  # Last 6 rows
str(df)                   # Structure
glimpse(df)               # Dplyr view
summary(df)               # Summary stats
dim(df)                   # Dimensions
nrow(df) / ncol(df)       # Rows / columns
names(df)                 # Column names
```

=== DATA CLEANING ===

```
df |> drop_na()           # Remove NAs
df |> drop_na(age)         # Drop NA in column
df |> distinct()          # Remove duplicates
df |> filter(!duplicated(.)) # Alt remove dup
replace_na(x, value)      # Replace NA
df$col <- NULL            # Remove column
df |> select(-col)        # Dplyr remove
```

=== DPLYR MAIN VERBS ===

```
df |> select(col1, col2)  # Choose columns
df |> filter(age > 30)    # Filter rows
df |> arrange(salary)     # Sort ascending
df |> arrange(desc(salary)) # Sort descending
df |> mutate(new_col = ...) # Create column
df |> summarize(mean_age = mean(age)) # Aggregate
df |> group_by(dept) |> summarize(...) # Group summary
```

=== VECTORIZED OPERATIONS ===

```
v <- c(1, 2, 3, 4, 5)    # Create vector
v[1]                     # First element
v[c(1, 3)]               # Elements 1 & 3
v[v > 3]                 # Conditional
v[-1]                    # All except first
v[which.max(v)]           # Max element
sort(v)                  # Sort
```

```
rev(v)                # Reverse
```

=== STATISTICS ===

```
mean(x), median(x), sd(x), var(x)    # Descriptive  
min(x), max(x), range(x)             # Min/max  
quantile(x)                          # Quantiles  
sum(x), prod(x)                      # Sum, product  
length(x)                           # Count  
unique(x)                           # Unique values  
table(x)                             # Frequency
```

=== HYPOTHESIS TESTS ===

```
t.test(x, mu=0)                  # 1-sample t-test  
t.test(x, y)                     # 2-sample t-test  
t.test(x, y, paired=TRUE)        # Paired t-test  
cor.test(x, y)                   # Correlation test  
chisq.test(table)                # Chi-square
```

=== LINEAR/LOGISTIC REGRESSION ===

```
model <- lm(y ~ x, data=df)       # Linear regression  
summary(model)                   # Model summary  
predict(model, newdata)           # Predict  
model <- glm(y ~ x, family=binomial) # Logistic
```

=== CLASSIFICATION MODELS ===

```
model <- rpart(y ~ ., data=df)    # Decision tree  
model <- randomForest(y ~ ., data=df) # Random forest  
model <- svm(y ~ ., data=df)      # SVM
```

=== MODEL EVALUATION ===

```
confusionMatrix(pred, actual)     # Classification  
mean(pred == actual)              # Accuracy  
accuracy_score(actual, pred)      # Caret  
rmse <- sqrt(mean((actual-pred)^2)) # RMSE
```

=== GGPLOT2 VISUALIZATION ===

```
ggplot(df, aes(x, y)) +  
  geom_point()                # Scatter  
  geom_line()                 # Line  
  geom_histogram(bins=20)     # Histogram  
  geom_boxplot()              # Boxplot  
  geom_bar()                  # Bar chart  
  facet_wrap(~var)            # Facets  
  labs(title="", x="", y="")  # Labels  
  theme_minimal()             # Theme
```

=== DATA RESHAPING (TIDYR) ===

```
df |> pivot_wider(names_from=x, values_from=y) # Long to wide  
df |> pivot_longer(cols=A:Z, names_to=x, values_to=y) # Wide to long  
df |> separate(col, into=c(a,b), sep=":")      # Split column  
df |> unite(col, a, b, sep="_")                # Combine columns
```

=== STRINGS (STRINGR) ===

```
str_trim(x)          # Remove whitespace
str_replace(x, "old", "new")  # Replace
str_to_upper(x), str_to_lower(x)  # Case conversion
str_length(x)        # Length
str_sub(x, 1, 5)      # Substring
str_detect(x, "pattern")  # Pattern match
str_split(x, ",")     # Split string
```

=== DATES (LUBRDATE) ===

```
Sys.Date()          # Today
ymd("2023-01-15")   # Create date
mdy("01/15/2023")    # Alt format
dmy("15-01-2023")    # Alt format
year(date), month(date), day(date)  # Extract parts
difftime(date1, date2, units="days")  # Date difference
```

=== FUNCTIONS ===

```
myfunction <- function(x, y) {
  result <- x + y
  return(result)
}
```

=== CONTROL FLOW ===

```
if (condition) {      # If
  # code
} else {
  # code
}
```

```
for (i in 1:10) {     # For loop
  # code
}
```

```
while (condition) {   # While loop
  # code
}
```

=== HELPFUL FUNCTIONS ===

```
?function_name        # Help
help.search("keyword")  # Search help
install.packages("pkg")  # Install package
library(pkg)           # Load package
sessionInfo()          # R info
set.seed(42)           # Reproducibility
getwd()                # Working dir
setwd("path")          # Change dir
```

=== SHORTCUTS ===

Tab: Auto-complete

Ctrl/Cmd + Enter: Run line/section

Alt/Option + -: Assignment operator (<-)

Ctrl/Cmd + Shift + M: Pipe operator (| >)

Ctrl/Cmd + L: Clear console

Ctrl/Cmd + K: Comment/uncomment